BLOCKCHAIN BASED SOFTWARE PROJECT INFORMATION SHARING AND ESTIMATION SYSTEM

BLOKZİNCİR TABANLI YAZILIM PROJE BİLGİSİ PAYLAŞIM VE KESTİRİM SİSTEMİ

MUSA ERHAN

ASSOC. PROF. DR. AYÇA KOLUKISA TARHAN

Supervisor

ASST. PROF. DR. ADNAN ÖZSOY

Co-Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering

2020

To my dear wife, Duygu...

and my son, Kerem..

ABSTRACT

BLOCKCHAIN BASED SOFTWARE PROJECT INFORMATION SHARING AND ESTIMATION SYSTEM

Musa ERHAN

Master of Science, Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Ayça KOLUKISA TARHAN

Co- Supervisor: Asst. Prof. Dr. Adnan ÖZSOY

January 2020, 92 pages

Accurate estimations play a significant role in the success of software projects, and companies should have sufficient number of past project data to make these estimations accurate and reliable. Some institutions gather project metrics from companies to create cross-company datasets and open these datasets to companies for paid or free of charge. On the other hand, many companies do not want to make public all or part of their project information so it prevents the growth of such datasets. Blockchain technology and smart contracts, as a medium to store private information and share it with predefined constraints, might be a solution to this problem. In this study, we propose a conceptual model as a reference for blockchain-based software project information sharing, and make a proof-of-concept implementation to discuss issues related to its feasibility. Then we make an example project estimation by using the proposed model. For this purpose, some project information in International Software Benchmarking Standards Group (ISBSG) [1] dataset is added to the system with the hand of a sample company. Then an analogy-based estimation is made by using project information in the proposed system. To the best of our knowledge, this is the first study to use blockchain within the context of software project information sharing and estimation.

Keywords: Blockchain, Software Project, Information Sharing, Access Control, Hyperledger Composer, Software Estimation

ÖZET

BLOKZİNCİR TABANLI YAZILIM PROJE BİLGİSİ PAYLAŞIM VE KESTİRİM SİSTEMİ

Musa ERHAN

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Danışmanı: Assoc. Prof. Dr. Ayça KOLUKISA TARHAN

Eş Danışman: Asst. Prof. Dr. Adnan ÖZSOY

Ocak 2020, 92 sayfa

Doğru kestirimler, yazılım projelerinin başarısında önemli rol oynar ve şirketler bu kestirimleri doğru ve güvenilir hale getirmek için yeterli sayıda geçmiş proje verisine sahip olmalıdır. Bazı kurumlar, şirketler arası veri kümeleri oluşturmak ve bu veri kümelerini şirketlere ücretli veya ücretsiz olarak sunmak için şirketlerden proje ölçümleri toplar. Öte yandan birçok şirket proje bilgilerinin tamamını veya bir kısmını kamuya açmak istememektedir ve bu durum, veri kümelerinin büyümesini engellemektedir. Gizli bilgilerin depolanması ve tanımlanmış kısıtlamalarla paylaşılması için bir araç olarak blokzincir teknolojisi ve akıllı sözleşmeler kullanılması, bu soruna bir çözüm olabilir. Bu çalışmada, blokzincir tabanlı yazılım proje bilgi paylaşımına referans olarak kavramsal bir model önerdik ve uygulanabilirliği ile ilgili konuları tartışmak için bir kavram-kanıt uygulamasını yaptık. Daha sonra önerilen modeli kullarak örnek bir proje kestirimi gerçekleştirdik. Bu amaçla Uluslararası Yazılım Kıyaslama Standartları Grubu (ISBSG) [1] veri setindeki bazı proje bilgilerini örnek bir firmanınmış gibi sisteme ekledik. Ardından, önerilen sistemdeki proje bilgilerini kullanarak analoji tabanlı kestirime bir örnek sunduk. Bildiğimiz kadarıyla bu çalışma, blockzincirini yazılım proje bilgisi paylaşımı ve kestirimi bağlamında kullanan ilk çalışmadır.

Anahtar Kelimeler: Blokzincir, Yazılım Projesi, Bilgi Paylaşımı, Erişim Kontrolü, Hyperledger Composer, Yazılım Kestirimi

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation and thanks to my supervisor Assoc. Prof. Dr. Ayça KOLUKISA TARHAN. Her expert guidance, supportive, encouraging and positive approach and her effort and patience throughout this time period made possible to complete this work.

I would like to express my thanks to my co-supervisor Asst. Prof. Dr. Adnan ÖZSOY for insightful comments and useful remarks during thesis study. His invaluable contributions helped my a lot in all the time of research and writing of this thesis.

I would like to thank my thesis committee members Asst. Prof. Dr. Tülin ERÇELEBİ AYYILDIZ, Assoc. Prof. Dr. Oumout CHOUSEİNOGLOU, Asst. Prof. Dr. Murat AYDOS, and Asst. Prof. Dr. Ömer Özgür TANRIÖVER for their time, useful comments and suggestions.

I would also like to thank TUBITAK BILGEM ILTAREN for supporting my graduate studies. I also want to thank to my colleagues for their encouragement and support.

I thank to my parents, for their love, patience, and every kind of support throughout my life.

I also want to thank my son, Kerem, who became the focus of my life, for his patience for the times I could not spend with him.

Finally, my deepest thanks to my wife, Duygu, who became the light of life with her love, endless patience, and support. Words cannot express how grateful I am for her being in my life. I cannot forget her altruism.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CLI	Command Line Interface
COCOMO	Contstructive Cost Model
DOM	Document Object Model
DoS	Denial of Service
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IFPUG	The International Function Point Users Group
IoT	Internet of Things
ISBSG	International Software Benchmarking Standards Group
JSX	JavaScript Extensible Markup Language
PoC	Proof-of-Concept
PwC	PricewaterhouseCoopers
QSM	Quantitative Software Management
REST	Representational State Transfer
SDK	Software Development Kit
SLIM	Software Lifecycle Management
SWIFT	Society for Worldwide Interbank Financial Telecommunication
UI	User Interface
XML	Extensible Markup Language

1. INTRODUCTION

Nowadays, as many software projects become larger and more complex, software estimation is needed more to achieve success. In the survey conducted by PWC on project management [2], poor estimation in the planning phase is the main reason (32%) for the failure of software projects. According to 2014 Standish Group Chaos Report [3], another important survey on this subject, 31.1% of the projects were cancelled and 52.7% of the projects was 189% more costly than original plan due to poor estimation. They estimated that in US, $81 billion is spent for cancelled software projects, and $59 billion is spent for software projects that cannot be completed on time.

Companies need past project data to establish software estimation practices and improve software project planning and management processes. A company can create its own within-company dataset from past projects. However, there are problems when relying on a within-company dataset [4]: (i) the company needs time to collect enough data on past projects; (ii) even if it has collected enough data in time, the company might have made changes on data for new projects, which could make their previous measurements not usable; and (iii) all data should be collected and kept consistently.

These problems have motivated the use of cross-company datasets. A cross-company dataset is a collection of project data that are collected voluntarily from several companies [5]. Some institutions aim to provide cross-company datasets by collecting project information from software companies either as part of a membership or for free. In case of membership, the company providing data can have access to the entire project database. Free access, similarly, allows all users to access project data gathered so far. However, this kind of access mechanism does not consider privacy issues neither on project nor attribute basis. As a result, companies that do not want to share all or part of their project information avoid data entry to these datasets.

In this work, with an aim to support creation of larger and trustable cross-company datasets, a conceptual model for blockchain-based software project information sharing is proposed. The conceptual model stores project information on blockchain and gives the owner of project information the authorization to determine access controls on the basis of project attributes. Basic features of blockchain technology such as data distribution, access-permission, and immutability have been considered in identifying operating principles underlying the conceptual model. For example, a company may add project information to the system with all attributes being accessible to third-parties, selected attributes being accessible to third-parties and others being private, or all attributes being private only for its own access. Accordingly, this company is assumed to make estimations more accurately based on its own project attributes at least, and based on other companies' project attributes as allowed for sharing by their owners in larger contexts. By this kind of access control mechanism, companies that do not use existing datasets for privacy reasons are expected to participate with the proposed model. It also provides an incentive mechanism to motivate the creation of larger cross-company datasets on which effective project estimations can be realized.

After defining the conceptual model, a proof-of-concept (PoC) implementation is made using Hyperledger Composer and deployed to Hyperledger Fabric Network. In order to evaluate blockchain-based implementation, a React web application is implemented. Finally, by using web application, the system is evaluated with provided operations which are importing and querying project information and making estimations with them.

The contributions of this thesis can be summarized as follows:

- Project data is stored on blockchain. With the features of blockchain, project information will be shared securely.
- An attribute-based access control mechanism is proposed. There is no attribute-based access control mechanism for storing and sharing project information in existing datasets. By using this mechanism, only data owner can manage the access controls of the project data. In addition, this mechanism brings the data owner the ability to make some part of project attributes private and the other part public.
- An incentive mechanism is proposed to encourage users to use the system. With this mechanism, it is aimed to increase the participation by receiving token payments from users who benefit from the system and gaining tokens to the users who benefit to the system.
- A conceptual model is proposed for sharing project information. By using proposed conceptual model, it is aimed to increase the number of project data to be added to the system and to create a project information dataset larger than existing datasets. With this conceptual model, it is intended to be a guide when designing an information sharing system using blockchain technology.

- An infrastructure is proposed for storing and sharing software project information using blockchain. Based on the PoC implementation, the proposed infrastructure is found to be feasible and adaptable for other kinds of information sharing and storage problems using blockchain.

The rest of this thesis is organized as follows: Section 2 provides background on several known software project datasets and the basics of blockchain technology, together with summary of blockchain and web application frameworks that are used in implementation steps. Section 3 gives a summary of related work that shed light to the creation of the conceptual model, and analogy-based estimation details that are implemented in this work. Section 4 explains the conceptual model and its elements, and demonstrates its operation over an example scenario. Section 5 gives detailed information on PoC implementation. Section 6 evaluates the proposed model over the PoC implementation for correctness, estimation efficiency and performance, and discusses the results. Section 7 concludes the work with highlights from this initial work and plans for future work.

2. BACKGROUND

2.1. Software Project Datasets

According to the study [6], Desharnais, ISBSG, and Contstructive Cost Model (COCOMO) datasets are widely used for software project estimations. Below are brief descriptions of these and another widely used QSM database.

- Desharnais [7][8]: The most commonly used publicly available dataset in the field of software effort estimation. It consists of 81 projects collected by J.M. Desharnais.
- ISBSG: There are different subscription options to access ISBSG dataset. It includes data for more than 9,000 IT projects.
- COCOMO [9][10]: COCOMO’81 is another publicly available dataset. It includes data 63 projects.
- QSM Software Project Database [11]: The QSM database has over 13,000 completed software project metrics. Access to QSM database is done through QSM SLIM Tools [12][13] provided by the owner of the dataset. To avoid identification of data owners, access to data can only be done in summary form which is the result of provided tools.

Although these datasets and some others have been subjected to numerous software project estimation studies in both literature and industry, they have some common drawbacks in storing and sharing data in general, as we list below:

- They do not consider different roles with respect to data (e.g., owner, verifier, and user) in storing and sharing of project information.
- There is no attribute-based access control mechanism for storing and sharing project information. Although all project information in many software project datasets (e.g. in ISBSG) is anonymized before it is added to the pool in order to prevent traceability of data owners, we cannot say that this is a decentralized access control mechanism because once the data is added to the dataset, the central authority has full control of the data. In QSM dataset, on the other hand, access to

the dataset is provided through tools to secure the data, and the complete control of the data is not in the data owner but in the dataset owner.

- The number of project entries is limited (except ISBSG and QSM datasets) because of the lack of access control mechanism mentioned above. Nevertheless, it is expected that project entries to ISBSG and QSM datasets will increase with the attribute-based access control mechanism that can only be managed by the data owner.
- In most cases, the previously shared project information cannot be withdrawn or closed to access in time.
- Once shared, the datasets are managed by third-party users and therefore, the reliability of data is restricted to the reliability of these users.
- On the basis of reliability mentioned above, there is no mechanism to prevent project information being tampered or hacked by an external user.
- There is no well-defined incentive mechanism to motivate project owners to share their project information.

The issues mentioned above highlight the need for a role-based, access-permissioned, and trustable infrastructure for software project information storage and sharing. We propose by this work that blockchain technology and smart contracts, as a medium to store private information and share it with predefined constraints, might be a solution to this need.

2.2. Software Effort Estimation

Software effort estimation is the process to predict the amount of effort needed to develop a software project. It improves project management. Many studies have been conducted on effort estimation models and many different approaches have emerged. As a result of these studies, the classification that is proposed by Boehm [14] shown in Figure 2.1 is one of the most recent classifications.

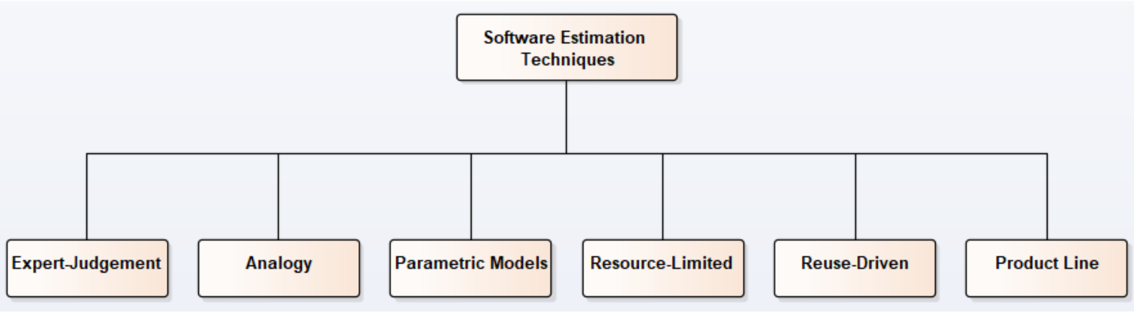


Figure 2.1. Boehm's Software Estimation Techniques Classification [15]

This classification differentiates estimation methods into six different techniques. The details of these techniques are listed below.

Expert-Judgement:

This type of model uses experts experience to estimate software project effort. It is one of the most widely used techniques. If the expert is not specialized in one part of project to be estimated and not specialized in other parts, this may cause errors in estimation. For this kind of cases, estimation is done with expert group to obtain more accurate results. One of the widely used model of this technique is Wideband Delphi [9] model. This model makes estimation by consensus of expert group with a coordinator.

Analogy-Based:

With this technique estimation results are obtained by comparing historical project data with project data to be estimated. The success of estimation is highly dependent on past project data. In order to increase the accuracy of estimation results, selection of project and project attributes should be done carefully. Also, adjustment mechanisms can be used. Case-Based Reasoning, which is one of the analogy-based estimation models, creates feature vectors from historical project data and produces estimation result by considering similar cases for the project to be estimated.

Parametric Models:

Parametric estimation models make estimations by performing mathematical relationships using statistical parameters. Effort estimation model is created with past project data and other parameters. COCOMO II [16], one of the most widely used models, includes applications composition, early design and post-architecture sub models. It uses the following formula.

$$Effort = A * Size^{B} * \prod_{i=1}^{N} EM_i \qquad (2.1)$$

Where

- Effort is in person-months
- A is a constant derived from past project data
- Size is in KSLOC (thousand source lines of code)
- B is an exponent dependent on additive scale drivers
- EM_i is an effort multiplier for the i^{th} cost driver.

Resource-Limited:

This type of model [16] defines cost or time that are independent attributes as limited resource and estimates are dependent attributes of project. One of the models using this technique, The Schedule as Independent Variable (SAIV), accepts the schedule as an independent variable, and accepts less relevant project attributes as dependent variables. Based on this assumption, it makes effort estimation over listed features.

Reuse-Driven:

This technique [16] uses phase information of projects with similar size and type to make effort estimation. With this technique, estimation is made by including the costs of the components and processes to be used in the project to be estimated.

Product Line:

Product line effort estimation technique [16] is used to estimate the effort of the project by including the costs of the components to be developed and the components to be reused in the estimation process during project development.

2.3. Analogy-Based Effort Estimation

As also mentioned in the previous subsection, analogy-based effort estimation is a technique that estimates the current project's effort using similar past project data. A certain number (k) of projects that have the closest analogies to the current project are selected from past projects. Then the current project's effort is estimated by adjusting the selected projects' effort.

Analogy-based estimation is the most suitable effort estimation method to evaluate the proposed system because of its highly dependent nature on past project data. In addition, it is a simple method and does not include intensive calculations. Therefore, the impacts of past project data can be identified easily.

In this study, analogy-based estimation and Mantel's test implementations are performed to test the PoC implementation of blockchain-based project information sharing infrastructure. Implementation details are explained in section 5.4.3 Mantel's Test and section 5.4.5. Analogy-Based Estimation. These implementations are made using the calculations described in the studies below.

Keung et al. [17] proposed an analogy-based estimation model named Analogy-X. It uses statistical calculations to detect relationship of input projects and project attributes. To detect and filter non-relationship inputs, Analogy-X runs Mantel's correlation.

The Mantel's test procedure starts with constructing two matrices (Figure 2.2) A and B;

$$A = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & b_{12} & \cdots & b_{1n} \\ b_{21} & 0 & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & 0 \end{bmatrix}$$

Figure 2.2. Mantel's Distance Matrices [17]

Each a and b value are calculated using euclidean distance with n project count and p project attributes. Before euclidean distance calculation, all values are transformed to be in the range of 0 and 1 by dividing max-min of project attribute value.

$$a_{21} = \sqrt{\sum_{i=1}^{p} (x1_i - x2_i)^2} \tag{2.2}$$

Mantel's correlation coefficient is

$$R_m = \frac{\sum a_{ij} b_{ij} - \sum a_{ij} \sum \frac{b_{ij}}{m}}{\sqrt{\left[\left\{\sum a_{ij}^2 - \frac{\left(\sum a_{ij}\right)^2}{m}\right\} \times \left\{\sum b_{ij}^2 - \frac{\left(\sum b_{ij}\right)^2}{m}\right\}\right]}} \tag{2.3}$$

Only one of the diagonal elements is used in this calculation because A and B matrices are symmetric. So m is calculated as

$$m = \frac{n(n-1)}{2} \tag{2.4}$$

Analogy-X uses Jack-knife method to detect an outlier project and project attribute. To detect outlier projects, the following procedure is used.

- Calculate Mantel's correlation coefficient R_i value for all projects by excluding i^{th} project.
- Calculate Jack-knife estimator value $\bar{R}$ with

$$\bar{R} = \frac{\sum_{i=1}^{n} R_i}{n} \tag{2.5}$$

- Calculate standard deviation S with

$$S^2 = \frac{\sum_{i=1}^{n} (R_i - \bar{R})^2}{n-1} \tag{2.6}$$

- Calculate leverage metric of i^{th} project with

$$LM_i = R_i - \bar{R} \tag{2.7}$$

- Calculate standard normal form of impact with

$$z_i = \frac{LM_i}{S} \tag{2.8}$$

- Analogy-X only wants to filter extereme data points so they use $|z_i| > 4$ indicator of abnormal data points. They do not want sensitive to individual cases. In this thesis, $|z_i| > 2$ indicator is used to make the outlier detection process more sensitive. If $|z_i|$ is greater than 2, i^{th} project is outlier.

Analogy-X excludes outlier projects and runs analogy-based estimation using remaining projects.

Tsunoda et al. [18] made an empirical evaluation of outlier deletion methods for analogy-based cost estimation. They evaluated Mantel's Correlation based deletion. They calculate analogy-based estimation with following procedures.

	Effort	*Size*	*Metric₁*	*Metric₂*	…	*Metricⱼ*	…	*Metricₙ*
$Proj_1$	y_1	fp_1	x_{11}	x_{12}	…	x_{1j}	…	x_{1n}
$Proj_2$	y_2	fp_2	x_{21}	x_{22}	…	x_{2j}	…	x_{2n}
…	…	…	…	…		…		…
$Proj_i$	y_i	fp_i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{in}
…	…	…	…	…		…		…
$Proj_m$	y_m	fp_m	x_{m1}	x_{m2}	…	x_{mj}	…	x_{mn}

Figure 2.3. Project Data [17]

In Figure 2.3 $Proj_i$ is i^{th} project, $Metric_j$ is j^{th} project attribute, x_{ij} is a project attribute value, fp_i is the development size like function point and y_i is the effort value.

- Calculate normalized value of x_{ij} with

$$x'_{ij} = \frac{x_{ij} - \min(Mertic_j)}{\max(Metric_j) - \min(Metric_j)} \tag{2.9}$$

- Calculate the euclidean distance of project with

$$\mathrm{Dist}(Proj_a, Proj_i) = \sqrt{\sum_{h=1}^{m} (x'_{ah} - x'_{ihh})^2} \tag{2.10}$$

- Calculate the adjusted effort $adjy_i$ with

$$adjy_i = y_i \times \frac{fp_a}{fp_i} \quad (2.11)$$

- Calculate the estimated effort using adjusted efforts of k neighborhood projects (Simprojects) with

$$\hat{y}_a = \frac{\sum_{h \in Simprojects} adjy_h}{k} \quad (2.12)$$

To evaluate accuracy of estimated values MRE (Magnitude of Relative Error), MER (Magnitude of Error Relative to the estimate), and BRE (Balanced Relative Error) evaluation criteria are used. MRE means error relative to actual effort. MER means error relative to estimated effort. BRE means unbiased value of MRE and MER. Lower of these values means higher estimation accuracy. These values are calculated as the following where x is actual effort and $\hat{x}$ is estimated effort.

$$MRE = \frac{|x - \hat{x}|}{x} \quad (2.13)$$

$$MER = \frac{|x - \hat{x}|}{\hat{x}} \quad (2.14)$$

$$BRE = \begin{cases} \frac{|x - \hat{x}|}{x}, x - \hat{x} \geq 0 \\ \frac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} < 0 \end{cases} \quad (2.15)$$

2.4. Blockchain

Blockchain [19] is a distributed database that provides encrypted transaction tracking. It is invented by Satoshi Nakamoto whose real identity is still unknown. In blockchain, each record is digitally signed and a combination of records form so called ‘block’. Each block keeps the hash of the previous block, making a connected chain of blocks. This connected structure in blockchain avoids any alteration in the records, thus making it immutable. Its distributed nature gets rid of the need for a central authority for processing any transactions. So, the operations can be carried out directly between the buyer and the seller safely. Blockchain is a way for users to agree on something even if they do not trust each other. Figure 2.4 shows how blockchain works.

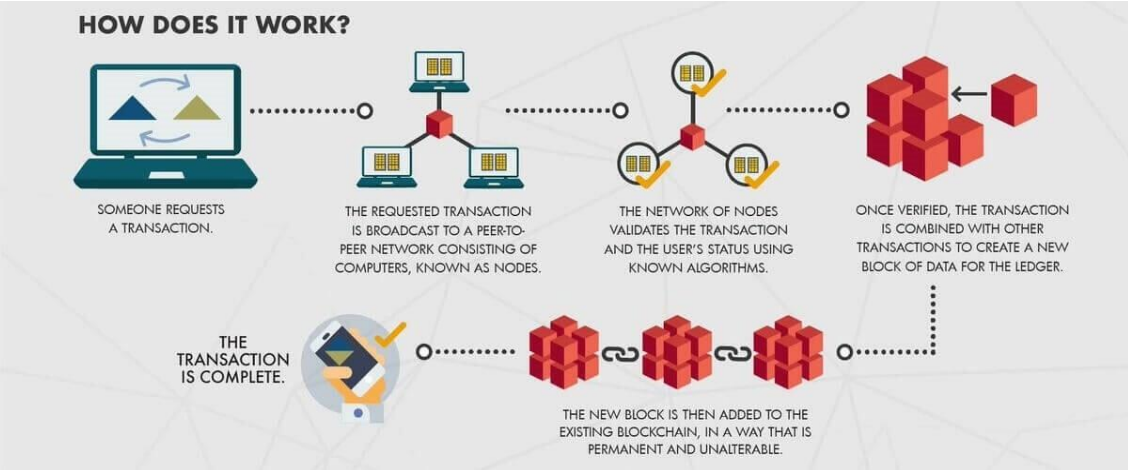


Figure 2.4. A Look at Blockchain [20]

As shown in Figure 2.4, someone requests a transaction for example sending bitcoin to another. The transaction is broadcast to nodes. Transactions are validated by nodes. A new block is created with some verified transactions. When the new block is added to existing blockchain, the transaction is completed.

Below is the list of main features of a blockchain:

- *Immutability*: Once a block added on the chain, it cannot be altered. So, it prevents corruption.
- *Decentralization*: A copy of the current information in the blockchain network is stored in different nodes. Blockchain does not need a trusted authority. Since there is no trusted authority, there is no one with ultimate rights to change the blockchain data for their own benefit. So only users who own the data can manage their data.
- *Security*: Blockchain security is based on cryptographic features such as asymmetric encryption and hash functions.
- *Transparency*: The history of the records can be followed by everyone.

There exist two major types of blockchain technology which are public and private. With a public blockchain anyone can access the network. But for a private blockchain there are restirictions to access the network. In addition to the main features that are mentioned above, both types have different features. Below is the list of the benefits of public blockchains:

- *Open Read/Write*: Anyone can submit transactions to the blockchain and can view all data related to transactions.
- *Distributed Ledger*: Each node is equal so the blockchain is immutable and censorship free.
- *Secure*: Anyone can be a node and contribute to the security of the system. With a lot of nodes in the network, it is much harder to attack the system.

The benefits of private blockchains are:

- *Faster Transactions*: Private Blockchain nodes distribute locally. This makes the performance faster.
- *Scalability*: Main scalability issues are related to consensus algorithms. But there are a number of fast consensus algorithms especially for private blockchains.
- *Member Control*: Only approved participants can submit transactions, and non-approved users cannot access to the blockchain. Therefore, no extra operations like encryption is required to prevent unauthorized users from accessing data.
- *Energy Consumption*: There are many consensus mechanisms with private blockchains that achieve consensus by consuming less resources.

Bitcoin [19] is the first and most popular digital currency that uses public blockchain technology. Sending bitcoin operations takes place in peer-to-peer network. It is faster especially compared to the SWIFT international money transfer system. Since there is no central authority, it is not under the control of any institution, organization or person. Bitcoins are created using the processing power in the distributed network that is called *mining*.

Later in 2015, Vitalik Buterin proposed Ethereum [21] that is a public blockchain-based computing platform. It enables the development of decentralized software protocols using its own special language. It has a cryptocurrency called Ether. Ether production is carried out by Ethereum miners. While cryptocurrency transfer was the main operation within the Bitcoin network, Ethereum aims to be a distributed computing environment in which users could integrate software applications on blockchain along with making cryptocurrency transfer. Ethereum Virtual Machine [21] is the infrastructure that runs programs called *smart contracts* on the Ethereum. Smart Contracts [21] are programs that can run automatically on the blockchain and work to meet certain conditions. Smart contracts are lines of code that are stored on blockchain. When a set of defined rules are met, smart contract automatically runs and produces results.

Hyperledger [22] is a Linux Foundation open source project that provides a variety of projects for building private blockchain networks for business. It primarily focuses on creating distributed ledger for institutions and business networks. For this purpose, different systems and tools have been developed in order to adapt to changing needs. Hyperledger is definitely not a cryptocurrency like Bitcoin and Ethereum. Smart contracts can be defined to execute logic that generates new facts that are added to the ledger like Ethereum.

Tokens represent an asset or benefit on project ecosystem. Owners can use tokens to access a service. The cryptocurrencies such as Bitcoin and Ether are tokens, but every token does not have to be a cryptocurrency. Some tokens are designed to be used within the market created by the application in which they are related.

2.5. Hyperledger Fabric

Hyperledger Fabric [23] founded by Linux Foundation under the Hyperledger Project umbrella is open source private blockchain technology. It is designed to solve different types of problems with its modular architecture. Hyperledger Fabric components are listed below:

Membership Service Provider:

Membership Service Provider is a component that authenticate users who want to access the network. This process can be carried out in different ways. (i) An identity list can be created and used with authorized users, (ii) identities produced by a Certificate Authority (CA) known by Hyperledger Fabric may be allowed, (iii) these two methods can be used together.

Peer:

Peers are the structures that make up the network and the ledger. All peers hold the ledger, and some run smart contracts. Hyperledger Fabric contains endorsing peer and committing peer. Endorsing peer simulates and endorses the transaction that comes from client. Committing peer validates and commits the transaction that comes from ordering service.

Orderer:

The transactions written to the ledger must be sequential. Hyperledger Fabric performs this sorting operation with the ordering service created by the nodes called orderer. Hyperledger Fabric supports two ordering mechanisms. (i) SOLO contains a single orderer node. This node sorts transactions by their times. (ii) Kafka contains orderer node group. These nodes order transactions by communicating each other. Other mechanisms can be implemented and added to Hyperledger Fabric modular system.

Channel:

Channel structures can be created on Hyperledger Fabric network. Groups can be crated between users who can access to the same network with channel structure. And it is possible to create a ledger only for these groups.

Ledger:

As we can see in Figure 2.5, ledger includes world state and blockchain components. World state keeps the current status of the ledger. Blockchain keeps information of all transactions that make changes to the world state.

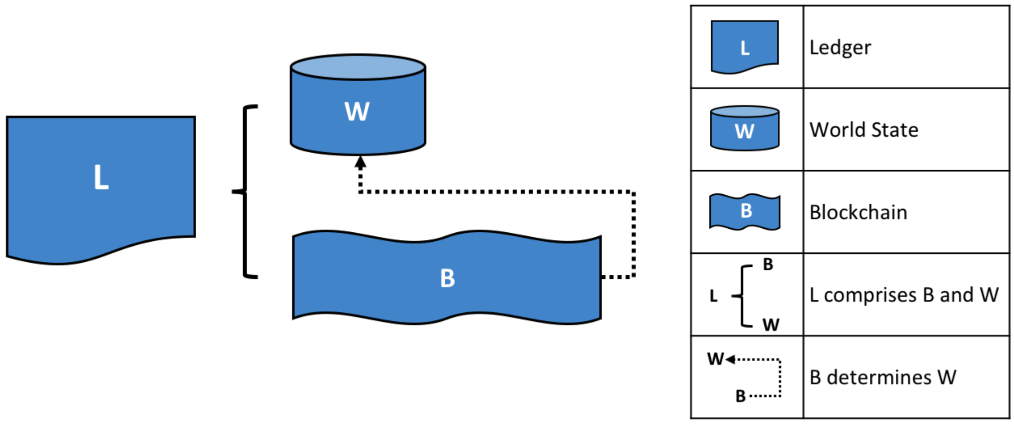


Figure 2.5. Hyperledger Fabric Ledger [24]

Chaincode:

Smart contracts are named as chaincode in Hyperledger Fabric network. Chaincode is an application that manages the state of the ledger and can work on the network. Hyperledger Fabric supports chaincode implementation with Go, Java and Nodejs programming languages.

Transaction:

As we can see in Figure 2.6, the transaction from client goes through the following processes before committing to peers:

- Endorsement: A transaction first comes to the endorsing peers. Endorsing peers control client signature and simulate the transaction. Endorsing peers create endorsement signature and return to the client. The client collects the endorsement signatures and sends them to the ordering service when signatures count is enough. This number is defined in the network endorsing policy information.
- Ordering: Ordering service sorts incoming transactions according to the specified ordering mechanism and sends them to the committing peers. Committing peers validate transactions. Adds transaction blocks to the blockchain and updates the world state.

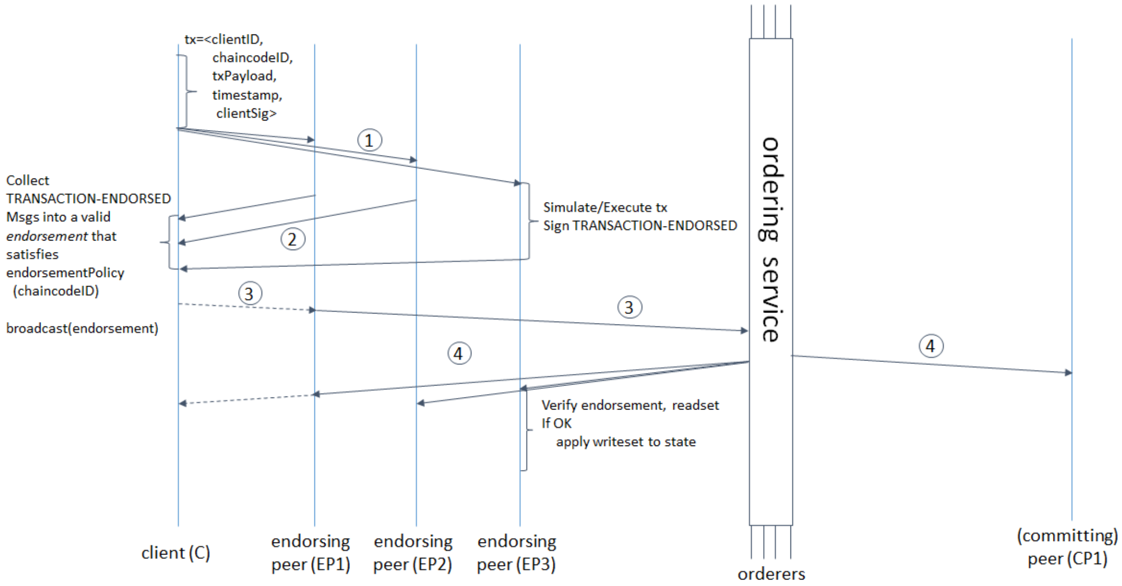


Figure 2.6. Hyperledger Fabric Transaction Flow [25]

2.6. Hyperledger Composer

Hyperledger Composer [26] is an open source blockchain framework. The aim is to create proof-of-concept blockchain applications (named as business network) by implementing blockchain application ideas quickly and easily. We can model our blockchain application with the following concepts provided by Hyperledger Composer:

- Assets: Assets are entities held in the business network. These can be anything. It includes the properties of an entity and may be associated with other assets or participants.

```
asset Toy identified by toyId {
  o String toyId
  o String name
  o Double price
}
```

- Participants: Participants are users defined in the business network. They can manage assets and call transactions.

```
participant Buyer identified by buyerId {
    o String buyerId
    o Double balance
}
```

- Transactions: It is a mechanism by which participants call and manage assets. For example, a customer call the buy transaction for a toy. If the customer has enough money, the required money is charged from the customer and the owner of the toy is the customer.

```
/**
* A transaction processor function description
* @param {org.market.BuyTransaction} tx Buy transaction
* @transaction
*/
function BuyTransaction (tx) {
  //Buy toy asset.
}
```

- Queries: It is a language defined to query assets, participants on blockchain network. It may contain basic filtering and sorting keywords.

```
SELECT org.market.Toy WHERE (price < 30)
```

- Access Control: Controls over participants can access to assets in business network. It is implemented by access control language which is provided by Hyperledger Composer.

```
rule BuyerCanAccessToys {
  description: "Buyer can access toys"
  participant: "org.market.Buyer "
  operation: ALL
  resource: "org.market.Toy
  action: ALLOW
}
```

Business network definition is created by implementing the concepts defined above. The business network definition is turned into a business network archive (Figure 2.7) to deploy to the Hyperledger Fabric network.

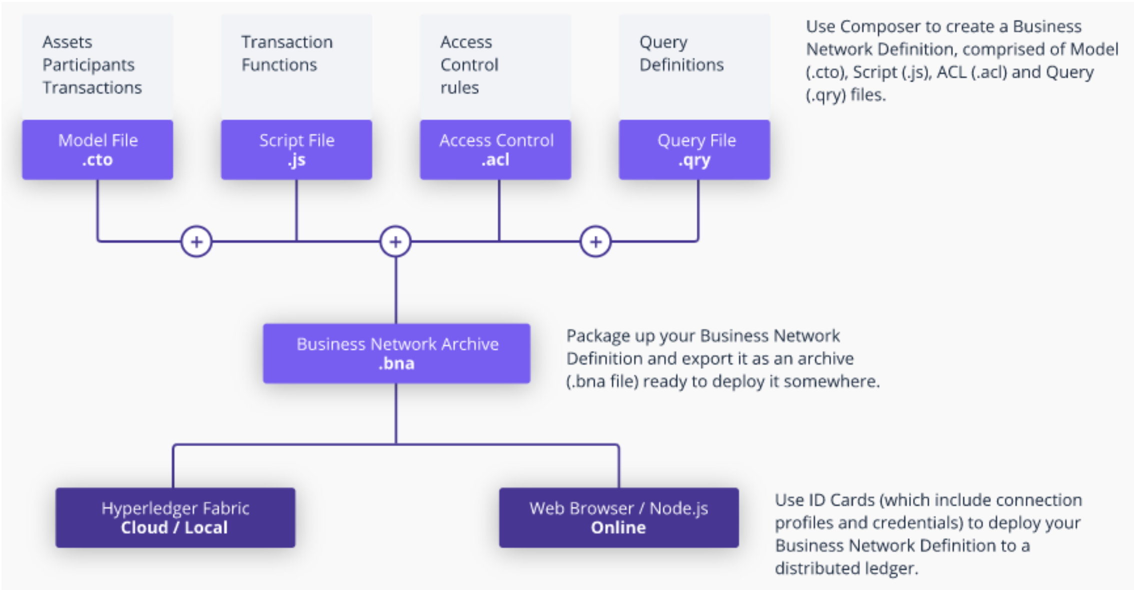


Figure 2.7. Hyperledger Composer Business Network Archive [27]

Other concepts used by Hyperledger Composer are as follows.

- Connection Profiles: Keeps the information necessary to connect to the Hyperledger Fabric network.
- Identities: It is created by mapping a user who wants to access the network and a participant existing in the network. A user accesses the network as a participant with created identity.
- Business Network Cards: It contains identity, connection profile and metada information about the network to be connected. It simplifies the process of connecting to the business network.
- Events: It is defined in the same way as assets and participants. A transaction can emit an event. And client appications can register events and receive emitted data.
- Historian Registry: Successful transactions are recorded in the historian registry. Historian record is also defined as an asset in Hyperledger Composer.

The full-stack architecture given by Hyperledger Composer as an example is shown in Figure 2.8.

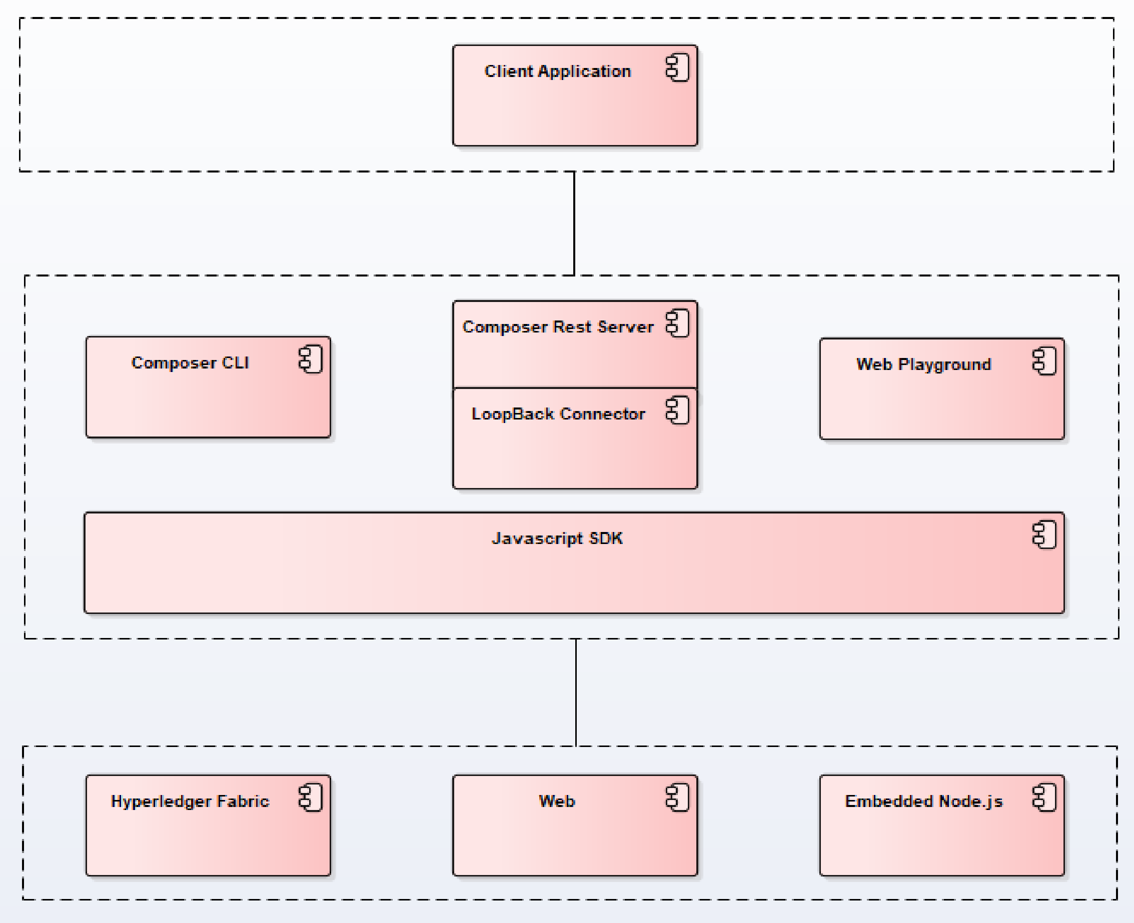


Figure 2.8. Hyperledger Composer Full-Stack Architecute [28]

Hyperledger Composer supports three runtimes:

- Hyperledger Fabric network. When this runtime is used, data is stored on Hyperledger Fabric ledger.
- Web, which is used by Playground. Data is stored in browser storage with web runtime.
- Embedded Node.js, which is used for testing purposes. Data is stored in-memory with embedded Node.js runtime.

Target runtime for business network card is specified in connection profiles. JavaScript SDK includes Node.js APIs. They are used to manage deployed business networks. Composer CLI is used to deploy and manage business network definitions. Composer Rest Server is used to generate REST API for business network. It provides create, read, update and delete operations for assets and participants. It also enables submit transactions. LoopBack Connector is used by Composer Rest Server. Web Playground is a web application to create a business network easily. Client Application is an application that access business network from Composer Rest Server.

2.7. ReactJS

ReactJS [29] developed by Facebook is an open source JavaScript library used to create UI components. A React component manages part of the UI. They are combined to create the desired UI. It aims to define and reuse independent components that make up the whole UI.

DOM is an interface for HTML documents to manage document structure and content. With DOM one can add, update ande delete document items that are accessed via nodes and objects. Changing an element over DOM requires traversing over DOM tree so this reduces performance. React implements virtual DOM to increase performace of DOM management. When the state of a component changes, virtual DOM is updated. Virtual DOM updates DOM with the most efficient way. For example, instead of applying each change individually, it applies all of them in a single pass. This improves performance by reducing repaint and reflow operations.

JSX can be used during React implementation. It is a JavaScript extension with an XML sysntax. It is used to create React elements. JSX codes are rendered as virtual DOM. It provides intention about how the appearance will be while managing UI elements with JavaScript code.

2.8. jQWidgets

jQWidgets [30] is a UI framework that provides components for mobile applications and websites. The components developed by jQWidgets using HTML5, CSS and JavaScript technologies can be used with different frameworks and libraries such as Angular, Vue and React. It supports many operations like component handling, event handling and property change notifications etc. Its aim is to help developing productive and fast UIs.

3. RELATED WORK

There are a lot of studies about blockchain and effort estimation, but not both as in the context of this thesis. Due to the lack of these studies, related work about blockchain are mentioned in the context of data sharing with privacy, and related studies about effort estimation are mentioned in terms of the impact of the number of included projects on effort estimation performance.

3.1. Blockchain-Based Data Sharing with Privacy

Blockchain data sharing with privacy has been studied by a number of researchers in recent years. We provide an overview of their related works below.

Azaria et al. [31] proposed a system called Medrec to handle medical record management using blockchain technology in 2016. With this system they aim to manage authentication, confidentiality, accountability and data sharing for, sensitive medical information. They use Ethereum and smart contracts to log patient information.

In 2016, Cruz et al. [32] proposed an authentication mechanism system suitable for the trans-organizational utilization of roles. Their system makes role-based access control using Bitcoin protocol. They designed a challenge-response authentication protocol to verify ownerships of roles.

Xia et al. [33] proposed another system that provides medical data sharing in cloud repositories called MeDShare. The system uses smart contracts and access control mechanism for all actions on data. They claim that sharing medical data with re-search and medical institutions with data privacy can be ensured by this system.

In 2018, Cruz et al. [34] proposed a role-based access control mechanism using Ethereum and smart contracts. Their mechanism verifies users by using a challenge-response authentication protocol. They compared their mechanism with other mechanisms based on smart contacts and Bitcoin blockchain.

In 2018, Liu et al. [35] proposed a data sharing agreement protocol which uses blockchain. Their protocol creates smart contracts based on agreement protocol and shares data in exchange for payment. Their framework includes a voting mechanism that can impose penalties. Their framework can be used for different kinds of terms associated with data sharing agreement.

In 2018, Ozyilmaz et al. [36] proposed a blockchain-based Internet of Things data marketplace using Ethereum and smart contracts. They used Swarm [37] as the distributed storage platform. They aimed to make IoT device vendors and Artificial Intelligence and Machine Learning solution providers work together.

In 2019, Kabi et al. [38] proposed a physical goods marketplace application using Ethereum. Their application enables trading of goods without a third-party. They measured the performance of the system based on gas which is a unit for computing power to execute an operation in Ethereum Virtual Machine.

3.2. Impact of the Number of Included Projects on Effort Estimation Performance

Within the scope of this thesis, we provide an overview of related works about improvement to the estimation accuracy with the number of included projects.

In 2014, Idri et al. [6] performed a comprehensive systematic mapping and review of analogy-based software development effort estimation. The authors examined 65 studies based on approaches, contributions, techniques used in combination with analogy, performance comparison with other estimation models, frequently used tools, and the impact on estimation accuracy of combining analogy with another technique. This work includes the studies (e.g., [39][40]) about increasing estimation performance by increasing the number of projects, which supports the proposal of this thesis. These studies are described in the following paragraphs.

In 2008, Mittas et al. [39] proposed data sampling method that uses nearest neighbor non-parametric regression to improve the performance of analogy-based effort estimation. Their results show that increasing the number of projects with their method reduces the error in analogy-based estimation.

In 2008, Kamei et al. [40] proposed another data sampling method that uses an over-sampling method to improve the performance of analogy-based effort estimation. Their evaluation results show that estimation accuracy improved by increasing the number of projects with their sampling method.

In 2008, Keung et al. [17] proposed an analogy-based estimation method called Analogy-X that uses Mantel’s correlation randomization test. They used Mantel’s test for project attribute selection and to detect outlier projects to improve analogy based estimation performance. Within the scope of this thesis, Mantel's test is included in the PoC implementation to detect outlier projects.

The studies aiming to increase the estimation accuracy with data sampling described above ([39][40]) have achieved improvements in estimation performance. Kamei et al. achieved 3.1% improvement for the Mean Magnitude of Relative Error (MMRE) accuracy measure. They also achieved 10.7% improvement for the MMRE for projects with large effort. As a result of evaluations made in different datasets, Mittas et al. achieved improvement between 11.35% and 27.77% for the MMRE accuracy measure.

3.3. Alternative Solutions

There is no alternative solution that provides all the features of the proposed system. For this reason, QSM dataset in terms of effort estimation and ISBSG dataset in terms of data sharing can be examined as alternative solutions. Users cannot directly access QSM dataset. The main purpose is to make estimations on the dataset by using QSM SLIM tool. The main purpose of ISBSG is to create a software project dataset and help companies improve their software project processes by using this dataset for a fee. The proposed system aims to increase the benefits of these two solutions with blockchain and attribute-based access control mechanism. It is expected that more project data will be collected with the proposed system by addressing the privacy issues of data owners. In addition, it is aimed to obtain reliable and accurate results with estimation methods to be integrated into the proposed system.

3.4. Contributions

By examining the existing software project datasets, it is seen that with the contributions of this thesis, a larger and more secure software project dataset can be created.

The first contribution is storing project data using blockchain. In this way, a secure storage environment for project data is created using the features of blockchain. Benefits of using blockchain to the proposed system are listed below.

- With decentralization feature, proposed model does not require a managing authority.
- Project data is stored in a distributed manner so there is no risk of losing data because all blockchain nodes will contain these data.
- With immutable feature, stored project information data cannot be tampered without the consent of the owner.

As the second contribution, an attribute-based access control mechanism is defined for software project data stored in the system. With this mechanism, which is not included in the existing software project datasets, all access rights of the project data are given to the control of the data owner. Thanks to the decentralization feature of the blockchain, the data provider trusts the system, not to an authority. Also, this mechanism motivates the data provider to participate.

The third contribution is the proposed incentive mechanism. With this mechanism, it is aimed to motivate to use the system by creating the win-win situation listed below for all roles of the system.

- To motivate data providers in sharing more project data, data providers earn tokens for shared project data as they are used by other data users.
- To motivate verifiers in verifying more project data, verifiers earn tokens as they verify project data in order to rate its reliability. This rating is then used to have an idea of the reliability of project data for data users.
- The data users pay tokens to access reliable project data as proportional to the amount used.

Although there is no real-world equivalent of token value within the scope of this thesis, it can be represented by a real-world asset in systems to be realized.

As the fourth contribution, a conceptual model is created for blockchain-based project information sharing. The conceptual model was tested with a sample scenario that was executed manually, and certain steps within this scenario (especially the ones related to the roles of “data provider” and “data user”) were realized in the PoC implementation. In future studies, when it is desired to develop a system that provides secure data sharing and access control using blockchain, the proposed conceptual model would be useful while assessing the needs and constraints of such a system and designing the system.

The fifth contribution is the proposed infrastructure for storing and sharing software project information using blockchain. The infrastructure is developed with Hyperledger Composer by using the conceptual model. In the PoC system developed within the scope of this thesis, the proposed infrastructure is implemented for software project information, but it can be adapted for other kinds of information storing and sharing problems using blockchain.

4. CONCEPTUAL MODEL

In this section, a blockchain-based conceptual model has been developed to investigate the feasibility of a role-based, access-permissioned, and trustable infrastructute for software project information sharing [41]. We explain details of the model and demonstrate its operations over an example scenario.

4.1. Ideas Behind the Conceptual Model

Blockchain-based conceptual model has three roles as data provider, verifier, and data user. The model offers several benefits to these roles. The data provider gains tokens in exchange of project information. Verifier earns tokens to verify the added project information. By motivating data providers to add project information, it is assumed that a large amount of project information will be added to the system and the relia-bility of these data will be ensured by the verification operations. The data user will have access to a large amount of reliable data generated by this model and in return he/she will pay token for the project information used.

The proposed model has an attribute-based access control mechanism for sharing project information. In this way, a data owner can manage access controls on the basis of project attributes and share all or part of its project information. The data provider can also manage previously shared project information and its access controls. In addition, an incentive mechanism is proposed to motivate project owners to add project information to the system.

4.2. Benefits of Blockchain

In this regard, blockchain technology provides significant benefits for the proposed model. With decentralization feature of blockchain our model does not require a managing authority. It is guaranteed that only the data provider can control transactions of the data. This enables the data provider to trust the system not to any authority and motivates to participate.

Project information and access control data will be stored distributed on blockchain. So there will be no risk of losing data because all blockchain nodes will contain these data. We can fetch missing data from another node.

The immutable feature of blockchain ensures that the project information saved in the chain will not be tampered without the consent of the owner.

4.3. Model

Figure 4.1 shows the graphical representation of the blockchain-based conceptual model. We explain the concepts and their relations following the figure.

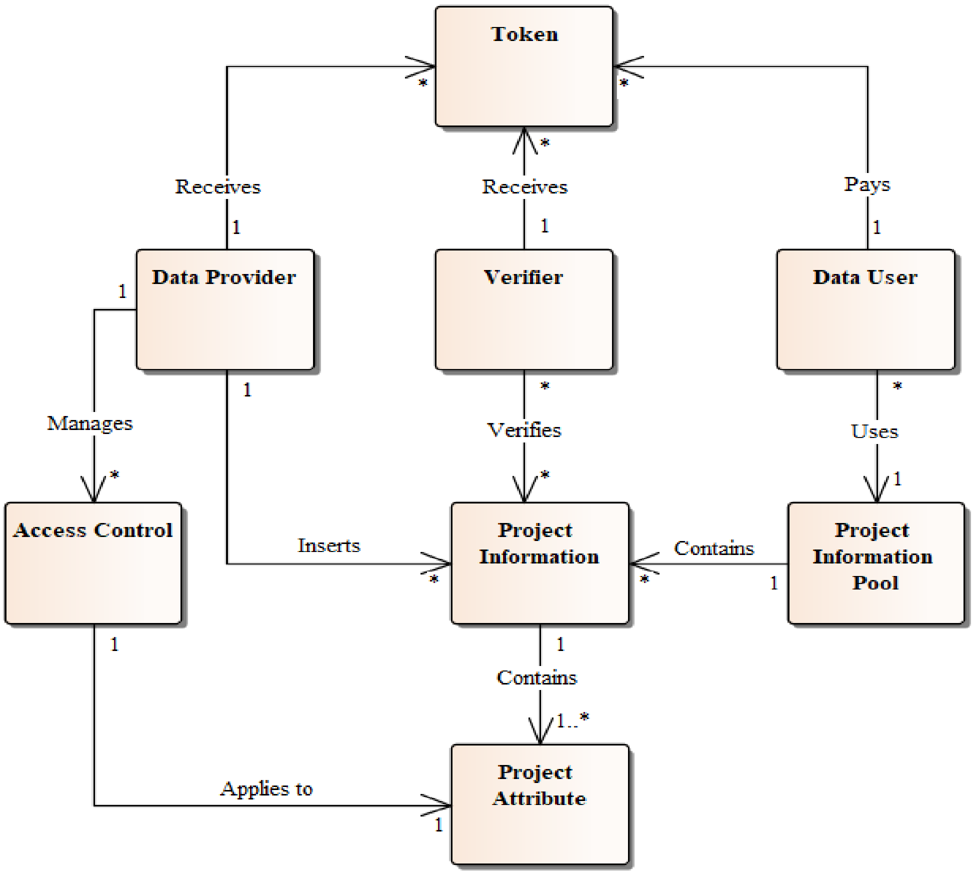


Figure 4.1. Conceptual Model for Blockchain-based Software Project Information Sharing

Data Provider: The data provider can insert project information to project information pool and manages access controls of already inserted project information. Access controls are managed by associating data users (or user groups to be defined) with project attributes that the data provider wants to grant access to. By this way, project information sharing is enabled per allowed attribute. Only the data provider owning project information is authorized to manage access controls. The data provider receives tokens in return for project information used by the data user. If the data provider does not give access right to anyone, he/she cannot earn tokens. Yet the data provider can access and use his/her own private project information.

Verifier: When the data provider adds project information, a number of data providers who have similar project attributes are assigned as verifiers. Similarity decision can be made by using project attributes like project type such as embedded system and project size measures such as functional size. Verifiers are selected from data providers who have been granted access to project information. Therefore, the verification cannot be done if the data provider does not give access to any other data provider. The reliability rating of inserted project information is determined according to the verification results. The value of rating which is verified by more verifiers will be higher. This value indicates the reliability of project information for data users. The verifier earns tokens after completing the verification process.

Data User: The data user makes queries in project information pool and uses project information that is granted access by data providers. Access to project information will be allowed on the project attribute basis. The data user pays tokens in exchange for using project information. The data user can evaluate the reliability of a project information according to its rating value. The project information which has higher rating value is more reliable because it is verified by more verifiers.

Project Information Pool: The project information pool is a collection of all projects' information. The information that data providers have added and data users have used is located in this repository. These data can be used in carrying out software project estimations by project managers.

Project Information: The project information defines the data of a software project. It contains project attributes of a software project.

Project Attribute: A project attribute is a piece of information that determines the properties of a software project. For example, a project attribute can be defined as type of software project which can be embedded system or mobile application. Access control of a project information is made on the basis of project attributes. In this way, a data provider can make a part of project information accessible and another part of project information private.

Access Control: Access control mechanism is used to enable data providers to add project information with the authorization restriction that they want. All access control management operations can be done by the data provider who is the owner of project information.

Token: The tokens are used to motivate data providers, verifiers and data users. It enables all roles to benefit from the system. The data provider is motivated to add project information. The verifier gains tokens by verifying project information. The data user will be able to access more reliable project information. It provides a win-win situation for all participating roles.

4.4. Example Scenario

Based on the conceptual model, the main flow of operations in sharing, verifying and using project information via the blockchain-based infrastructure is demonstrated in Figure 4.2. Numbers in the figure shows the order of operations in the flow.

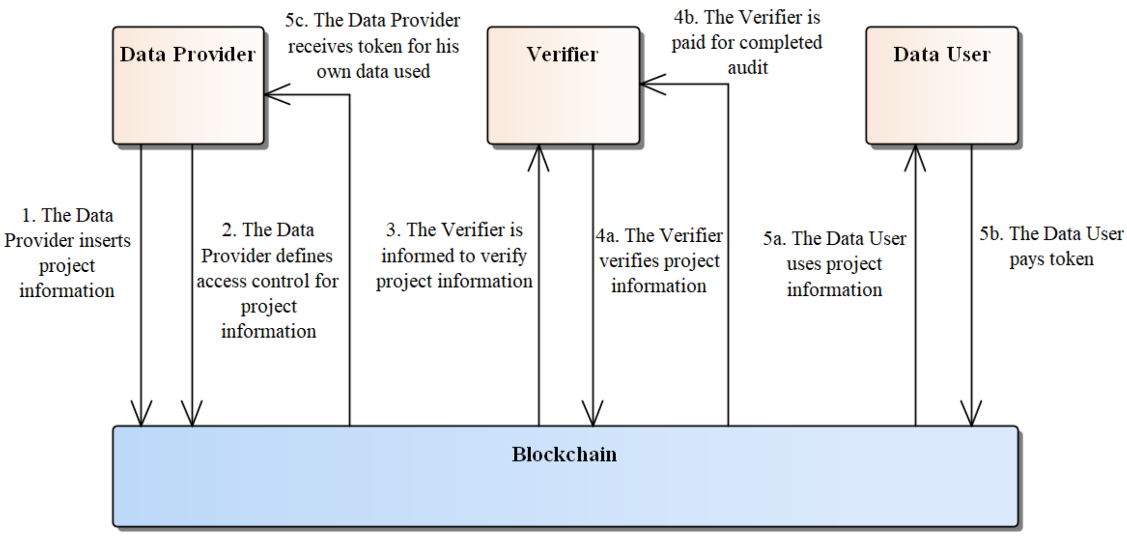


Figure 4.2. Main Flow of Operations in Sharing and Using Project Information

The following operations are executed in sequence in the main flow of the example scenario:

1. The data provider creates project information and sets project attributes. Then he/she inserts created project information into project information pool.

2. The data provider grants access rights for the project information via project attributes.

3. The verification request is made to inform the verifiers.

4. a. The verifier checks whether project information is proper and verifies it. Verification result determines the value of the project information rating.

b. The verifier receives tokens when the verification process is completed.

5. a. The data user queries project information pool with respect to certain criteria to obtain the project information data set that he/she want to use. Project information that the data user has access rights is displayed.

b. The data user pays tokens for using project information.

c. Token payment is made to the owners of project information used by the data user.

The scenario described above can be realized by developing a software application that will use the underlying blockchain technology. Smart contracts can be implemented for the data store and access control mechanism. The use-case diagram of such a software application to carry out the operations in the scenario is given in Figure 4.3 with respect to three roles defined as the actors.

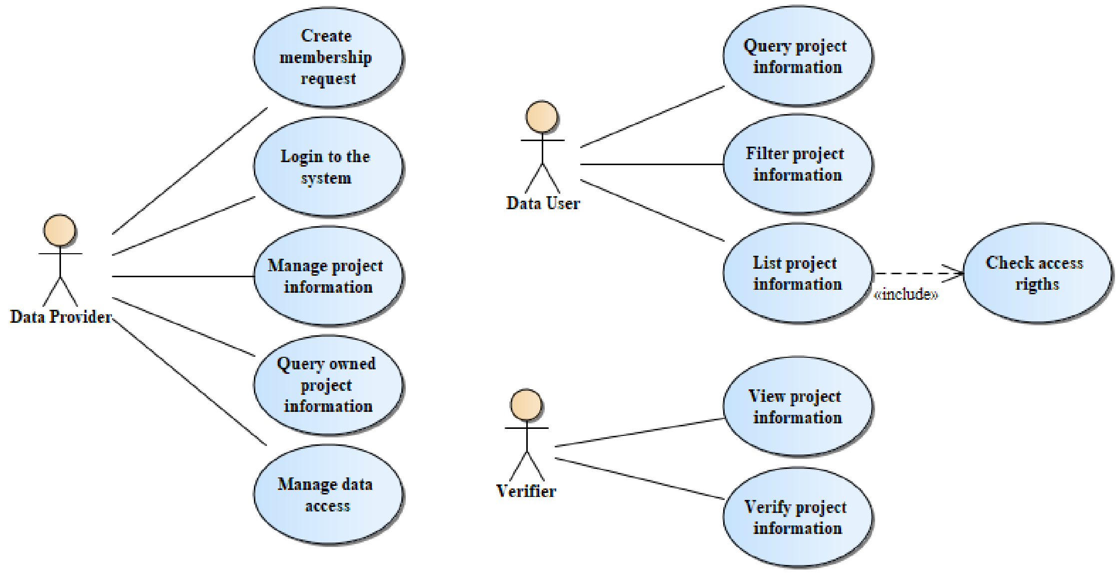


Figure 4.3. Use Case Diagram for Realizing the Example Scenario

4.5. Example Mapping

The proposed conceptual model is designed to store and share project information in different data sets. The mapping table given in Table 4.1 illustrates the use of the model. In the table, a sample part of ISBSG dataset in Figure 4.4 is adapted to the conceptual model.

Table 4.1. Concepts and ISBSG Dataset Mapping

Proposed Concept	**ISBSG dataset**
Project Attribute	A Cell (Ex: Manufacturing)
Project Information	A Row (Ex: Project information with ID 10132)
Project Information Pool	All Rows
Data User	Customer of Dataset
Verifier	Dataset Repository Manager
Data Provider	Data Owner of a Row
Token	None. Only paid membership for Data Users
Access Control	Centralized, repository-based

	Rating	*Rating*	*Software Age*	*Major Grouping*	*Major Grouping*	*Major Grouping*
ISBSG Project ID	Data Quality Rating	UFP rating	Year of Project	Industry Sector	Organisation Type	Application Group
10132	B	B	2010	Manufacturing	Manufacturing;	Business Application
10247	B	B	2009	Manufacturing	Manufacturing;	Business Application
10248	A	A	2012	Communication	Telecommunications;	Business Application
10249	A	B	2006	Government	Government;	Business Application
10273	B	B	2009	Manufacturing	Manufacturing;	Business Application
10323	A	A	2002	Insurance	Insurance;	Business Application
10332	B	B	2009	Manufacturing	Manufacturing;	Business Application

Figure 4.4. A Sample Part of ISBSG Dataset

As seen from Table 4.1, most of the proposed concepts can be mapped to those in existing ISBSG dataset. To better demonstrate the difference of the proposal with respect to the current operation, the following example changes can be made while mapping ISBSG dataset to the proposed model:

- The project information pool will include all rows but as allowed by data owner for a specific type of data user.
- Verifiers will be selected from data owners having similar project attributes.
- Role-based token will be defined to enable all roles to benefit from the system.
- Decentralized and attribute-based access control mechanism will be implemented with smart contracts on blockchain.

5. MODEL IMPLEMENTATION

In this section, proof-of-concept design and implemantation are explained in order to demonstrate that the proposed model is feasible and meets the basic requirements. First, general system architecture of the model is mentioned. Then the modules implemented in architecture are explained.

Proof-of-concept implementation supports operations 1, 2, 5.a, and 5.b in the operation flow described in Figure 4.2 in the previous section. In relation to this, from the use case diagram in Figure 4.3, "Manage project information" and "Manage data access" use cases of the data provider were implemented for basic data entry and use cases of the data user were focused more. Import, query, Mantel's test, analogy-based estimation, and estimation evaluation system scenarios are detailed at the end of this section.

Remaining operations in Figure 4.2 and use cases in Figure 4.3, which are not included in the scope of implementation, are predicted to be meaningful on a real system with a large network. In this thesis, approximately 3500 lines of code were written using 3 different frameworks for implementation. The time spent for implementation by including Blockchain, Hyperledger Fabric, Hyperledger Composer, Smart Contract, Reactjs, and jQWidgets learning curve was approximately 4 months. The scope of the implementation has been determined as described in the previous paragraph, taking into account the required effort within this thesis and the basic requirements of the model.

The development environment information used during system implementation is given below:

- Operating System: Ubuntu 16.04
- Blockchain: Hyperledger Fabric 1.1
- Blockchain Framework: Hyperledger Composer 0.19
- Smart Contract Development Language: Nodejs
- Web Application Javascript Framework: Reactjs
- Web Application UI Framework: jQWidgets

Development environment installation is described in detail in Appendix 1.

5.1. System Architecture

The architecture of the system is shown in Figure 5.1. This architecture is created by adding the components indicated in yellow to the typical solution architecture proposed by Hyperledger Composer (Figure 2.8). The Project Information Sharing Business Network that provides the blockchain operations needed by proposed model is deployed to Hyperledger Fabric. Since the web application, which is implemented for users to access the system, uses React JavaScript Library and jQWidgets User Interface Framework, these components are also included in the architecture.

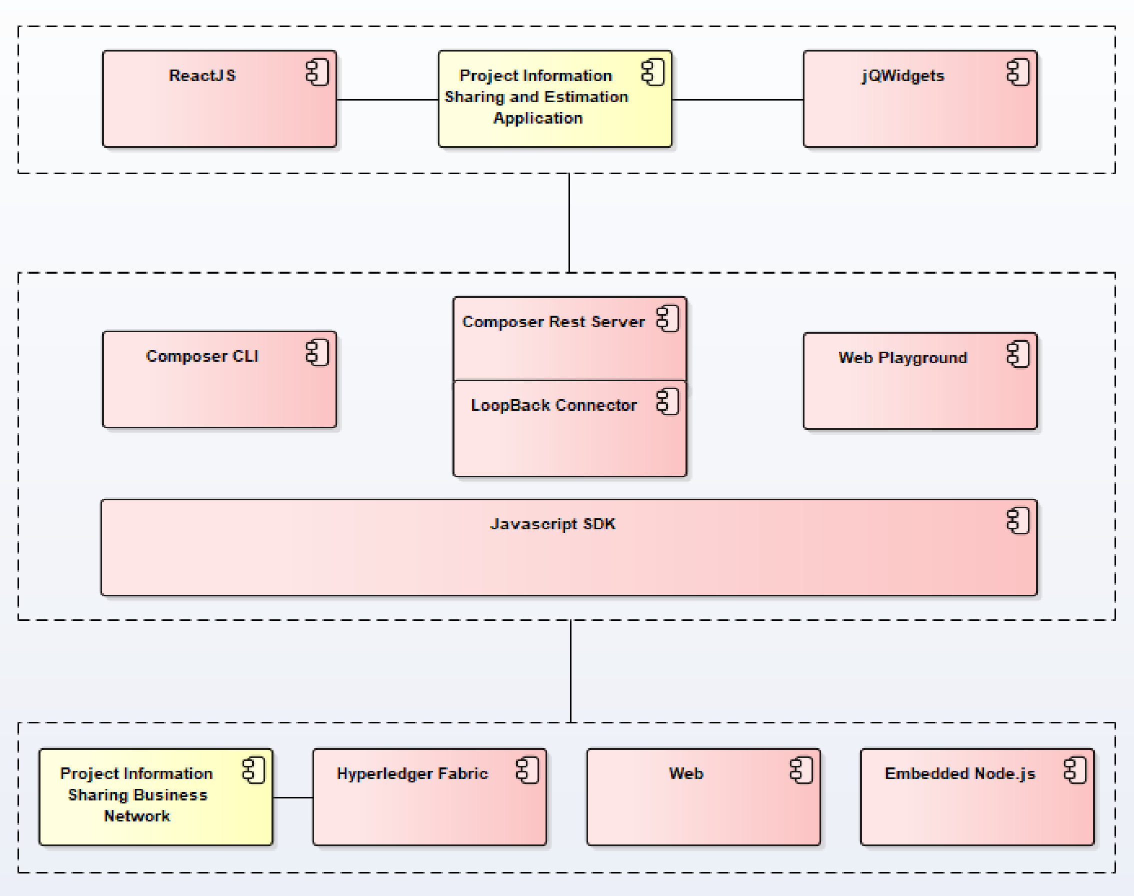


Figure 5.1. System Architecture

Project Information Sharing and Estimation Application makes blockchain access via Composer Rest Server. When Composer Rest Server receives an HTTP POST request for a transaction, it transmits this transaction message to the Hyperledger Fabric network using the JavaScript SDK. Hyperledger Fabric runs this transaction as implemented in Project Information Sharing Business Network that we deployed. The transaction results return to Composer Rest Server. And Composer Rest Server sends the results via HTTP Response to the requesting application.

Generating Composer Rest Server is described in detail in Appendix 2.

5.2. Project Information Sharing Business Network

The Project Information Sharing Business Network Definition component is implemented to provide all blockchain operations that the proposed model needs. With this component, a business network archive (.bna file) is created, and deployed to Hyperledger Fabric network. In this way, blockchain operations are performed over Hyperledger Fabric. The elements that make up this component are shown in Figure 5.2.

Deploying a business network archive is described in detail in Appendix 3.

Although the Project Information Sharing Business Network Definition is implemented as an infrastructure that enables storing and sharing software project information within the scope of the thesis, it is designed to be adaptable for other kinds of information storage and sharing problems using blockchain.

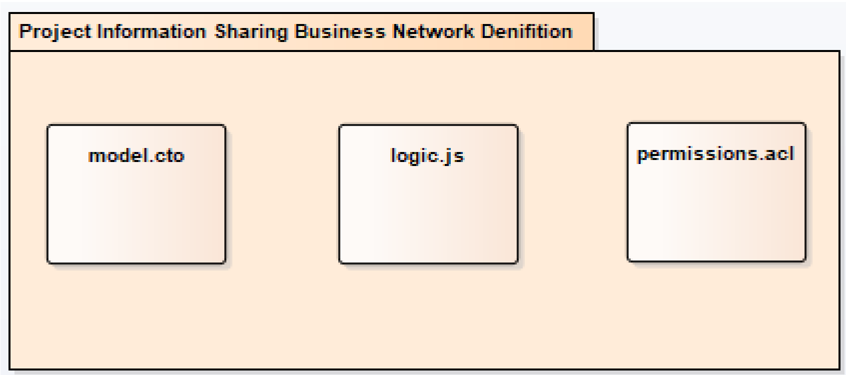


Figure 5.2. Project Information Business Network Definition

5.2.1. Model

The domain data model required for the business network is defined in the model.cto file using Composer Modelling Language. The following is a class diagram (Figure 5.3) showing the assets, participants, concepts, and transactions data and relationships defined in this file.

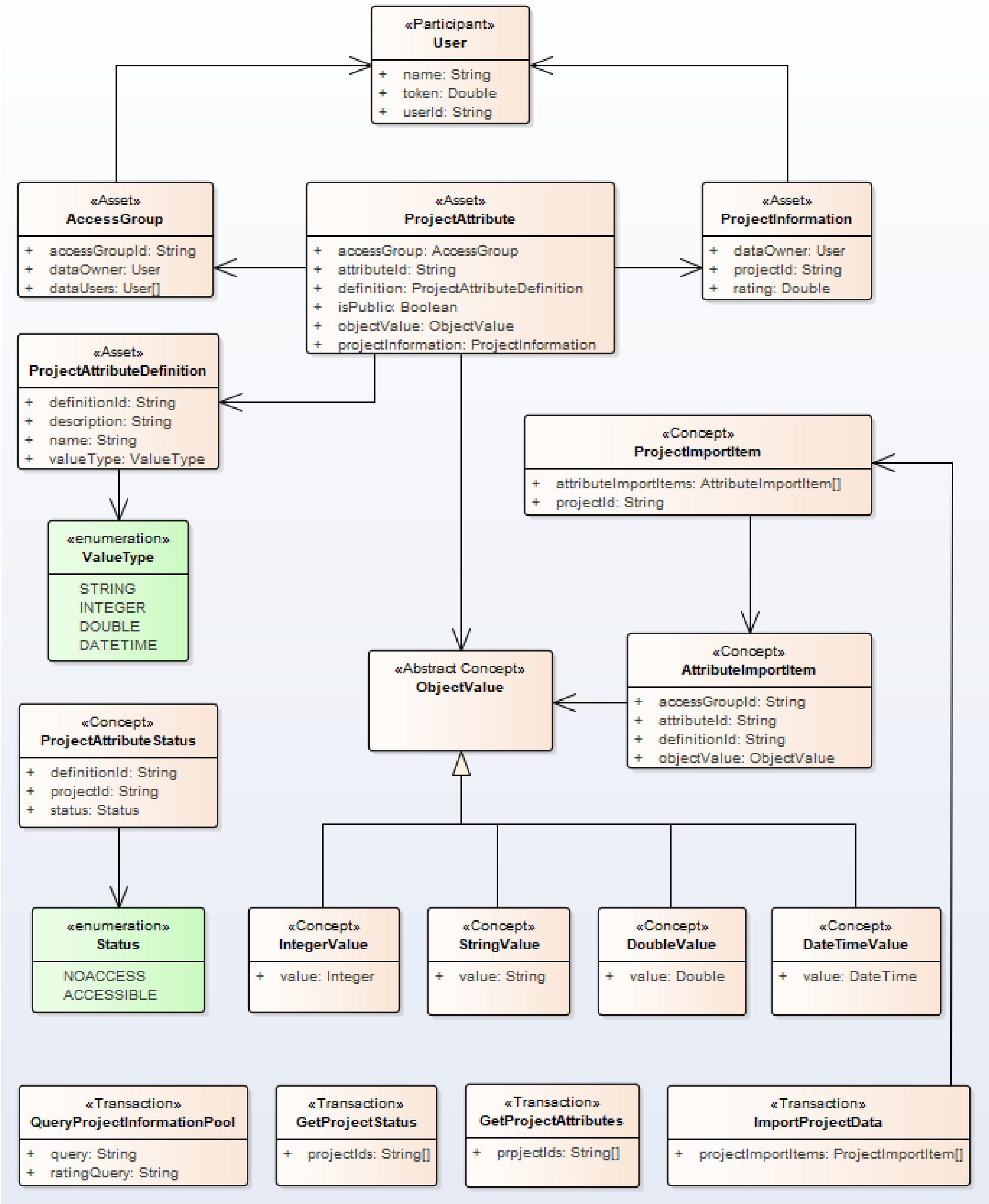


Figure 5.3. Project Information Business Network Definition Class Diagram

The classes shown in the class diagram are detailed below:

- User participant holds tokens information of a user to spend for loading project attribute.
- AccessGroup asset is created by a user and is assigned to a ProjectAttribute asset to give access rigths for users.

- ProjectInformation is an identifier of a project. It contains rating value of a project that is assigned by verification.
- ProjectAttribute is a piece of data that determines the properties of a software project. Data users pays for this data. It includes definition field to hold relation with ProjectAttributeDefinition asset. This relation is used to hold project attributes in a flexible and extensible way. Project attributes are stored in rows rather that columns and column information is taken from ProjectAttributeDefinition to make it easier to add new types of project attributes. Another important field of ProjectAttribute is accessGroup which holds the users who can access this project attribute. To store value of project attribute, objectValue relation is used for different kinds of data types. Hyperledger Composer does not support object type so ObjectValue concept is used like a polymorphic type. One child type of ObjectValue is used to store project attribute value such as IntegerValue, StringValue, DoubleValue, and DateTimeValue.
- ProjectAttributeStatus concept is used to show the project attributes that the current user can access.
- ProjectImportItem and AttributeImportItem concepts are used to import project data to the system.

The transactions are explained in the next section.

5.2.2. Logic

The transactions defined in model.cto file are implemented in logic.js file. The transactions are described below:

QueryProjectInformationPool: It filters project information according to query and ratingQuery parameters and returns results. The query string that comes as a parameter is used to filter the definition and objectValue fields in the ProjectAttribute asset. The other parameter, the ratingQuery string, is used for filtering through the rating field in the ProjectInformation asset. The results of transaction are ProjectInformation assets that match the filters. ProjectInformation asset is an identifier of ProjectAttributes and it does not contain any sensitive information. To facilitate users' access to the most appropriate project information, this transaction is performed without applying the access control rule and charging token from current user.

In this transaction, Hyperledger Composer API does not meet the filtering requirements, so Hyperledger Fabric Native API is used to perform filtering operations. Filter strings are prepared according to the format specified in Hyperledger Fabric Native API. Sample filter parameters are as follows.

{"query":"{\"$or\":[{\"$and\":[{\"definition\":\"resource:org.pis.ProjectAttributeDefinition#2002\"},{\"objectValue\":{\"value\":{\"$gt\":2015}}}]}]}","ratingQuery":"{\"rating\":{\"$gt\":85}}"}

GetProjectStatus: It checks whether the user calling the transaction has access to ProjectAttribute assets specified by given projectIds parameter. Then it creates ProjectAttributeStatus concept for each ProjectAttribute asset which is specified by projectIds parameter. And it sets status field with the current user's access rights. Since ProjectAttribute asset requires privacy concern, there is no information about ProjectAttribute at the end of the transaction. This transaction is used to show the project information which the user has access to, so no tokens are charged from the current user.

GetProjectAttributes: ProjectAttribute assets which are specified by given projectIds parameter are determined according to the current user's access rights. For ProjectAttribute assets that are determined, token is charged from the participant. If the charge operation is successful, ProjectAttribute assets are returned as a result. The token to be paid is determined by the number of Project Attribute assets. If the user does not have enough tokens, an exception is returned.

ImportProjectData: ProjectInformation and ProjectAttribute assets are created and saved by using the given project information and project attributes as parameters. Ids of saved ProjectInformation assets are returned as a result of the transaction. The owner of the imported project information is the current user.

5.2.3. Access Control List

All access controls required by the proposed system are provided by the access rules defined in the permissions.acl file. The following access rules are defined:

- **GetProjectAttributeDefinitionUserRule:** Allows that all participants can read ProjectAttributeDefinition assets.
- **AuthorizedProjectAttributeAccessRule:** Allows that data owner participant can read owning ProjectAttrbute assets.
- **GetProjectAttributesAccessGroupRule:** Allows that GetProjectStatus transaction can manage AccessGroup assets.
- **GetProjectAttributesTransactionRule:** Allows that all participants can create GetProjectAttributes transaction.
- **GetProjectAttributesUserRule:** Allows that all participants can call GetProjectAttributes transaction.
- **GetProjectStatusRule:** Allows that all participants can call GetProjectStatus transaction.
- **QueryProjectInformationPoolRule:** Allows that all participants can call QueryProjectInformationPool transaction.
- **ImportProjectAttributeRule:** Allows that all participants can call ImportProjectData transaction.
- **DefaultAccessRule:** Prevents all participants from accessing other resources.
- **ParticipantReadNetworkRule:** Allow that participants can access system information.

5.3. Project Information Sharing and Estimation Application

Business Information Sharing and Estimation Application is a web application that enables users to use the system. The blockchain operations are performed by communicating with the composer rest server through this application. While developing web application, React JavaScript library and jQWidgets JavaScript UI framework are used as shown in system architecture in Figure 5.1. The web application class diagram showing the classes and their relationships is given in Figure 5.4.

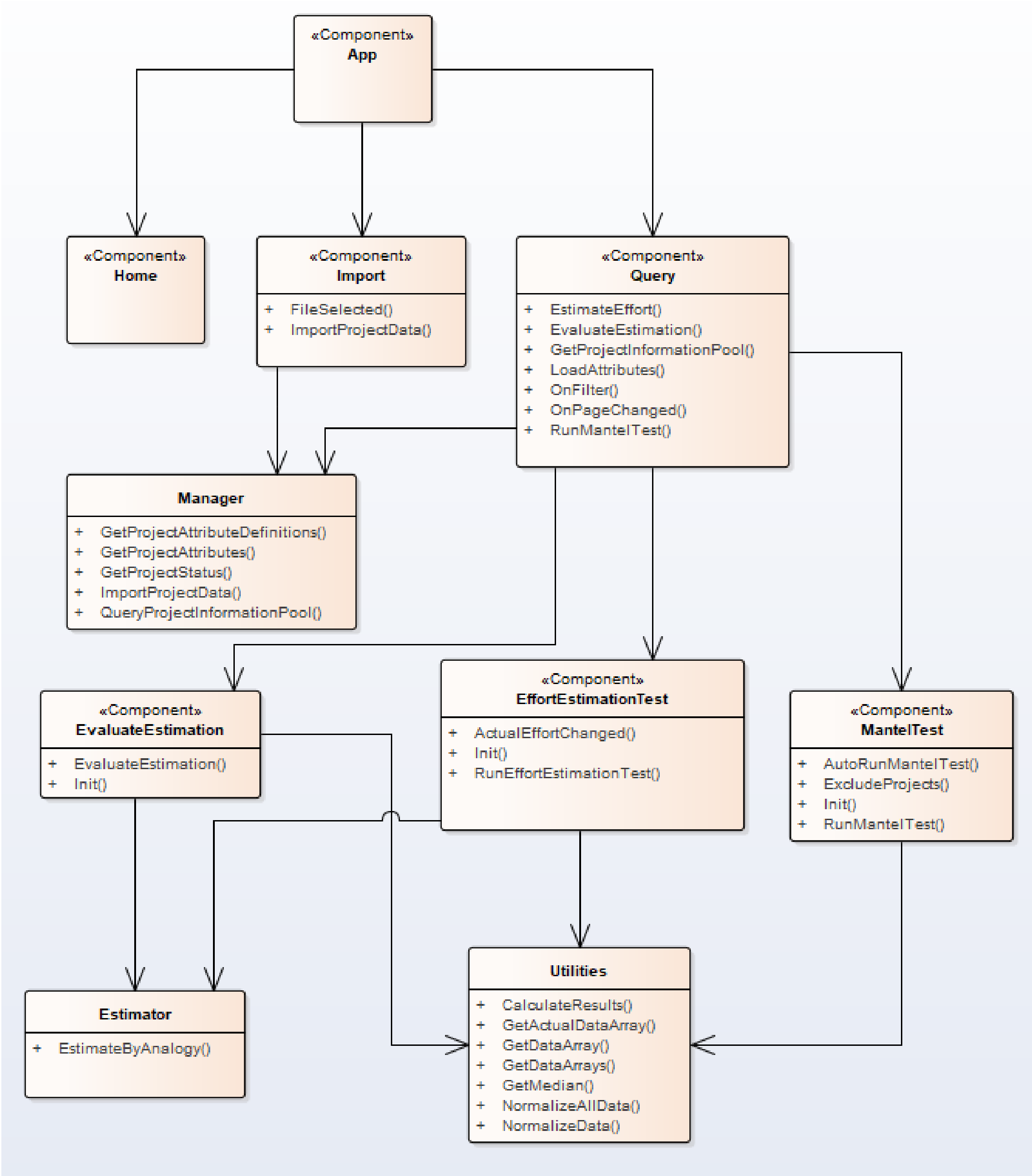


Figure 5.4. Business Information Sharing and Estimation Application Class Diagram

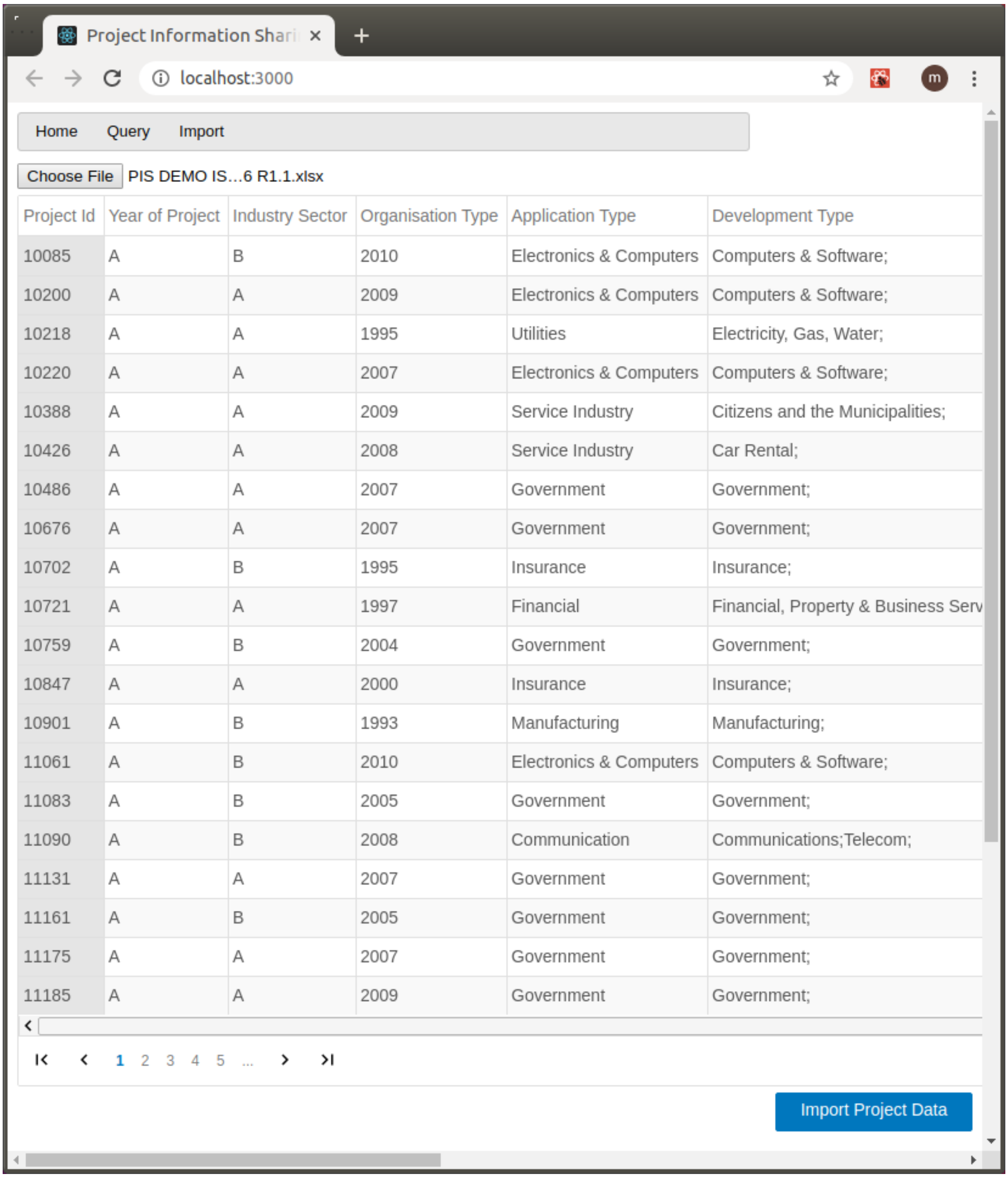
Information about the classes shown in the class diagram are listed below:

App Component: It is the main component of the application. Other components are included in this component. This component is rendered first when the application is opened. User can render Home, Import and Query components by using the menu component.

Home Component: It is the component that welcomes the user when the application is opened.

Import Component: It is the component used to import project data into the system. The following operations are performed on the page (Figure 5.5) rendered by this component.

- The file to be imported is selected with "Select File" button.
- Selected file is read.
- The project data that are read is imported to the system with "Load" button.



Project Id	Year of Project	Industry Sector	Organisation Type	Application Type	Development Type
10085	A	B	2010	Electronics & Computers	Computers & Software;
10200	A	A	2009	Electronics & Computers	Computers & Software;
10218	A	A	1995	Utilities	Electricity, Gas, Water;
10220	A	A	2007	Electronics & Computers	Computers & Software;
10388	A	A	2009	Service Industry	Citizens and the Municipalities;
10426	A	A	2008	Service Industry	Car Rental;
10486	A	A	2007	Government	Government;
10676	A	A	2007	Government	Government;
10702	A	B	1995	Insurance	Insurance;
10721	A	A	1997	Financial	Financial, Property & Business Serv
10759	A	B	2004	Government	Government;
10847	A	A	2000	Insurance	Insurance;
10901	A	B	1993	Manufacturing	Manufacturing;
11061	A	B	2010	Electronics & Computers	Computers & Software;
11083	A	B	2005	Government	Government;
11090	A	B	2008	Communication	Communications;Telecom;
11131	A	A	2007	Government	Government;
11161	A	B	2005	Government	Government;
11175	A	A	2007	Government	Government;
11185	A	A	2009	Government	Government;

Figure 5.5. Import Page

Query Component: It is the component used to query and use project data. The following operations are performed on the page (Figure 5.6) rendered by this component.

- The “project id” and “rating” values of the projects in the system are displayed together with the access status of the project attributes.
- Project data is filtered with column filter controls.
- The projects are selected with checkboxes in the first column.
- With “Load” button, the project attributes of selected projects are loaded and their values are displayed.
- The projects to be included in the estimation or Mantel’s test are selected using the checkboxes in the first column.
- With “Select Columns” button, the columns are selected to include in the estimation or Mantel’s test.
- “Mantel’s Test” button is used to open Mantel’s Test page, “Estimate By Analogy” button is used to open Analogy-Based Estimation page and “Evaluate Estimation” button is used to open Evaluate Estimation page.

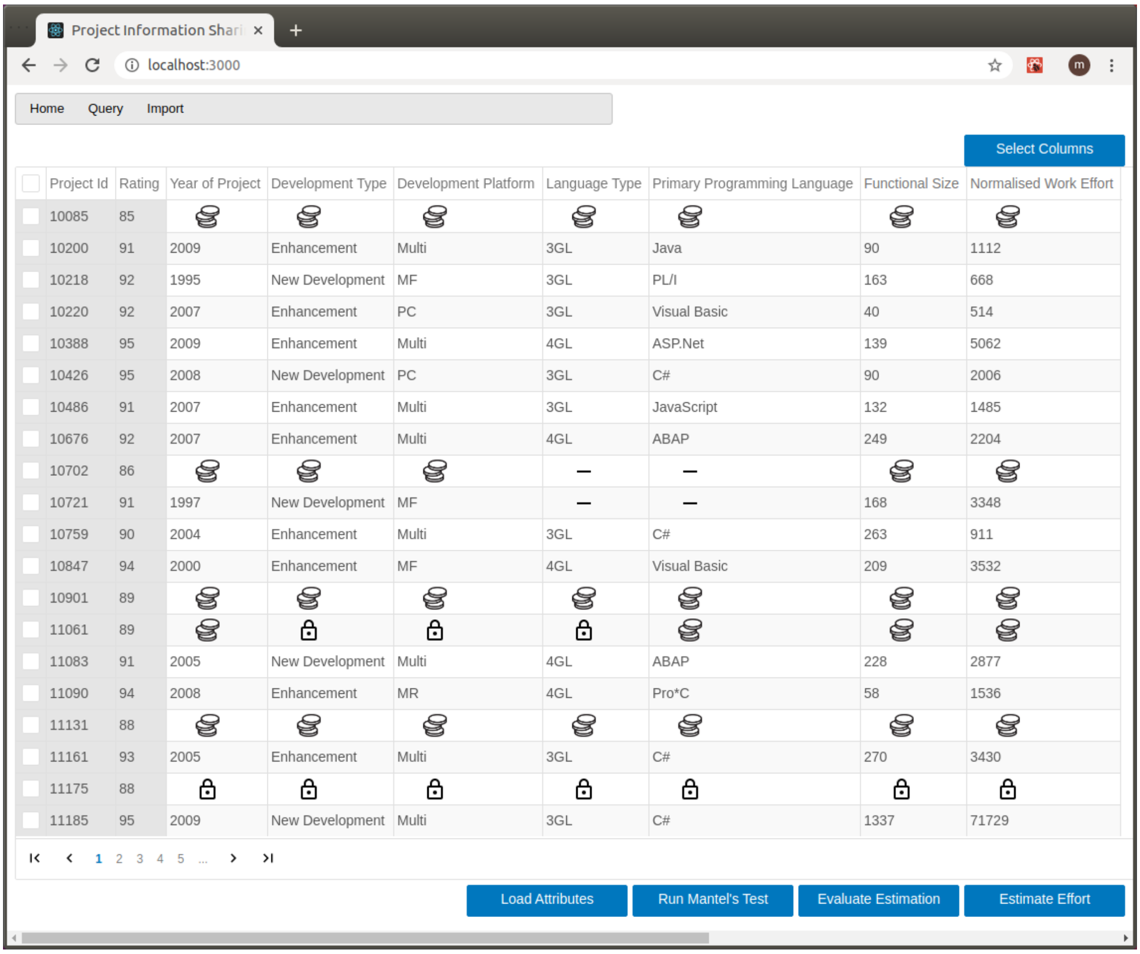


Figure 5.6. Query Page

MantelTest Component: It is the component used to run the Mantel's test. The following operations are performed on the page (Figure 5.7) rendered by this component.

- Dependent columns are selected with "Select Dependent Columns" combobox.
- Manual test is run with "Mantel's Test" button.
- After running manual test, projects that have $|z_i| > 2$ value are unselected with "Exclude Projects" button.
- With "Auto Run Mantel's Test" button, mantel test is run and projects that have $|z_i|>2$ are unselected until there is no project with $|z_i| > 2$ value.

  The completed test results are displayed.

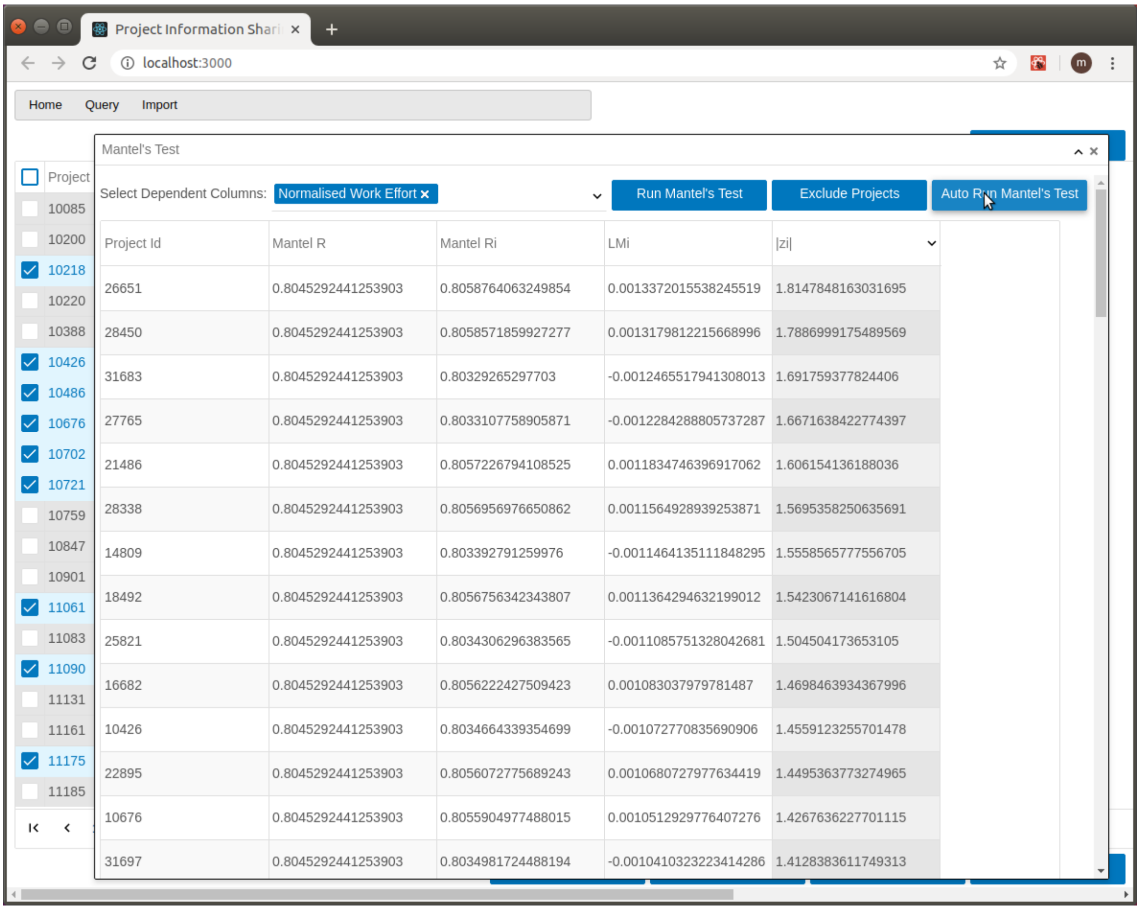


Figure 5.7. Mantel's Test Page

EffortEstimationTest Component: It is the component used to make analogy-based estimation. The following operations are performed on the page (Figure 5.8) rendered by this component.

- The effort column is selected with “Select Effort Column” combobox.
- The adjustment column is selected with “Select Adjustment Column” combobox.
- Analogy-based estimation is run with “Estimate By Analogy” button.
- The completed test results are displayed.
- By entering the actual effort value of the estimated project, statistical information results are displayed.

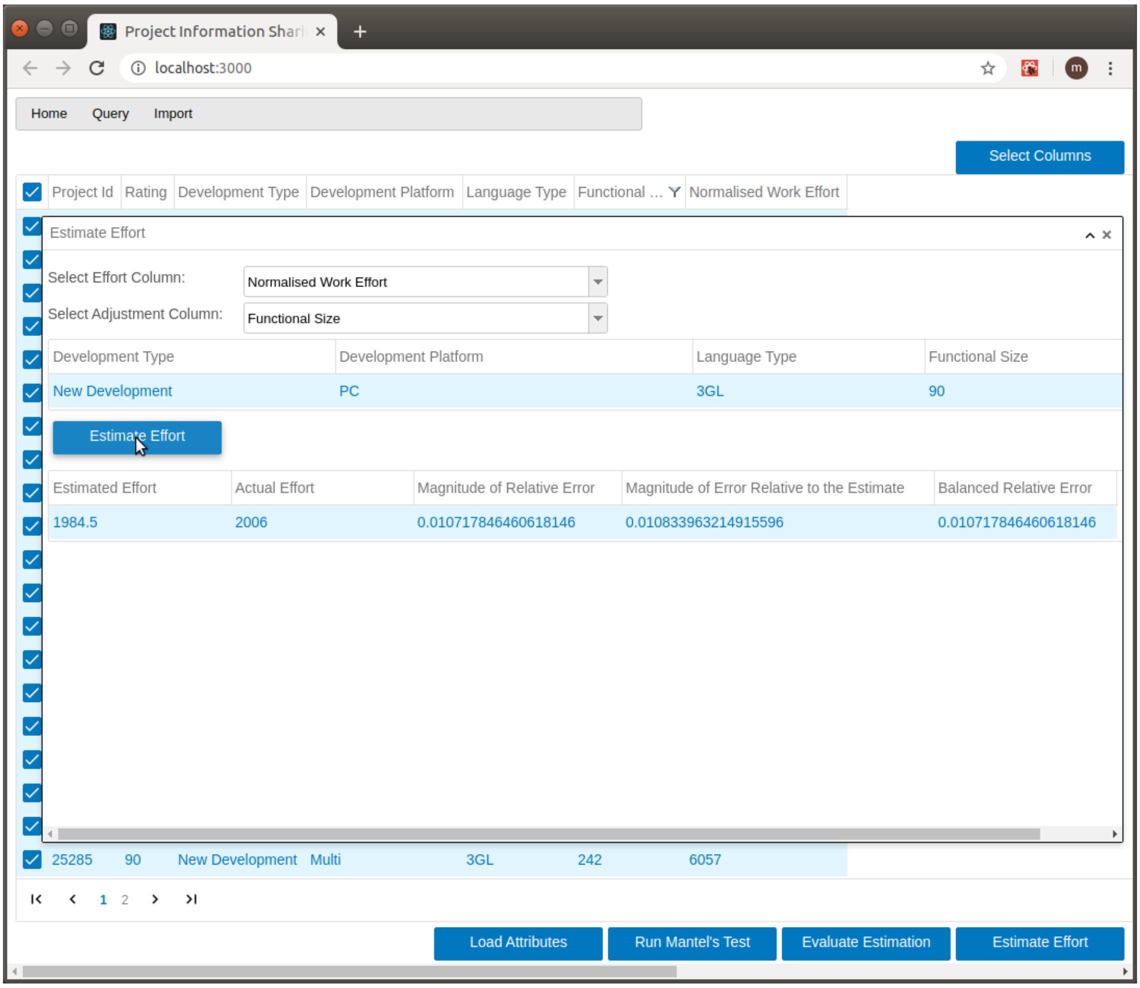


Figure 5.8. Analogy-Based Effort Estimation Page

EstimationEvaluation Component: It is the component used to evaluate analogy-based estimation. The following operations are performed on the page (Figure 5.9) rendered by this component.

- The effort column is selected with "Select Effort Column" combobox.
- The adjustmend column is selected with "Select Adjustment Column" combobox.
- With "Evaluate Estimation" button, analogy-based estimation is performed for each of the selected projects in the query page.
- Statistical results are displayed using the completed estimation result and actual effort information.

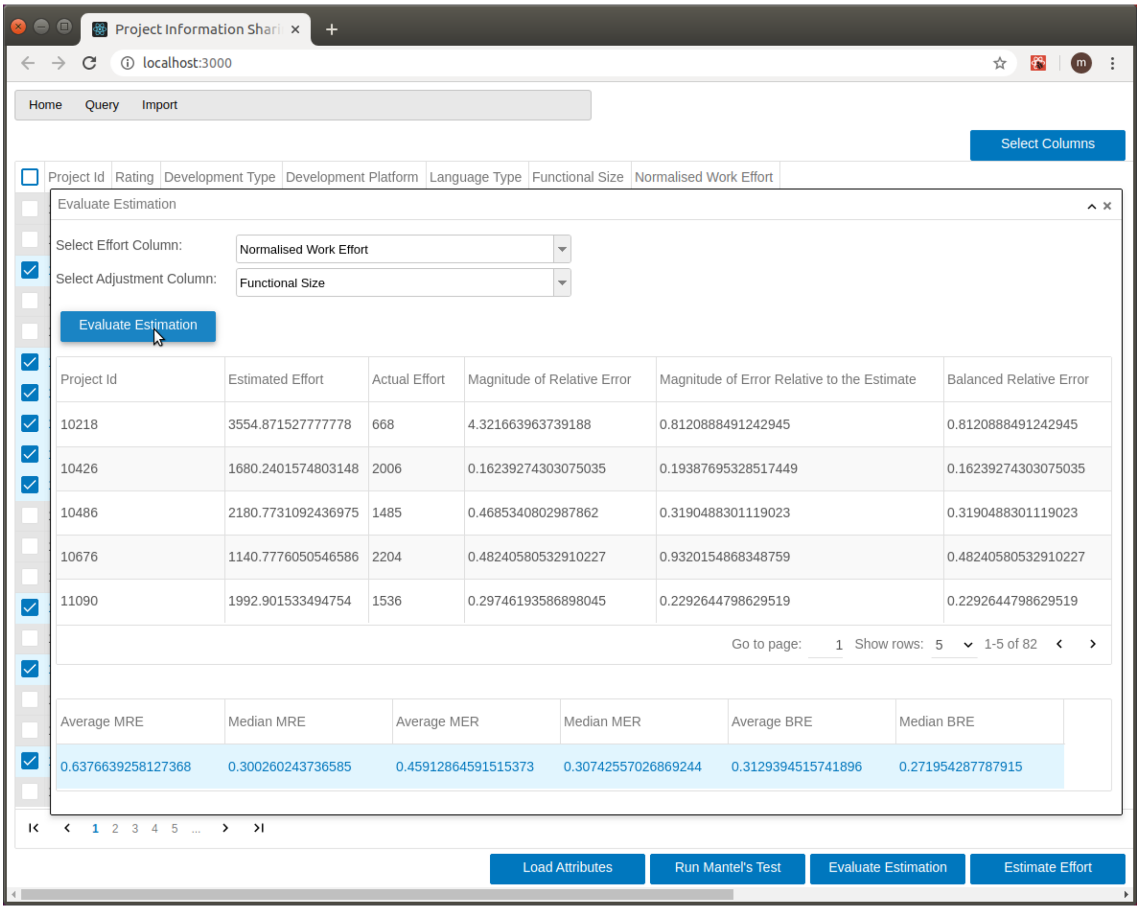


Figure 5.9. Estimation Evaluation Page

Estimator: It contains the calculation helper methods of analogy-based estimation. Components that make estimation use this helper class.

Utilities: It contains the calculation helper methods of estimation and Mantel's test.

Manager: A helper class that provides communications between components and composer rest server. All blockchain operations are made through this class.

5.4. System Scenarios

The following scenarios are implemented in order to meet proof-of-concept system's basic requirements. With these scenarios, the project data can be imported to the system, the project data stored in the blockchain can be filtered and effort estimation can be performed by using the project data that is granted access permission. It can be tested using these scenarios in which the system works correctly. The detailed description of these scenarios are given below.

5.4.1. Import

When the application is started, the import page is opened from the menu. “Select File” button is pressed. In the file dialog, the excell file containing the project data is selected. Project data is read from selected file and is displayed in the grid. “Load” button is pressed. Project data is imported to the blockchain. The user is notified when the import operation is completed. The sequence diagram showing the interactions of this scenario among the system components is shown in Figure 5.10. It includes operations 1 and 2 in Figure 4.2 and “Manage project information” and “Manage data access” use cases in Figure 4.3 with predefined rules.

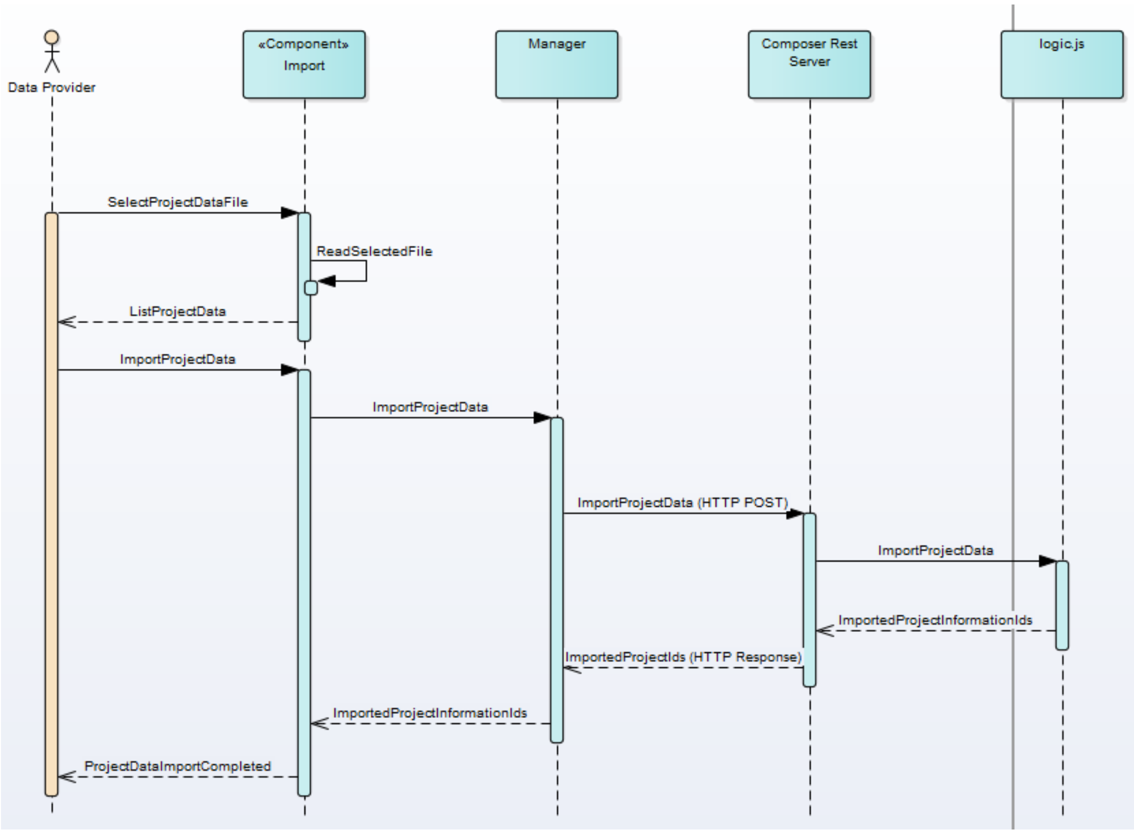


Figure 5.10. Import Project Data Sequence Diagram

5.4.1. Query

When the application is started, the query page is opened from the menu. When the page is opened; id, rating value and icons indicating access status of the user for each project attribute are displayed on the grid. Project data is filtered using the column filter controls.

The projects are selected to load project attributes and “Load Attributes” button is pressed. The tokens are paid for the corresponding project attribute values on blokchain and displayed in the grid. The sequence diagram showing the interactions of this scenario among the system components is shown in Figure 5.11. It includes operations 5.a and 5.b in Figure 4.2 and “Query project information”, “Filter project information”, “List project information”, and “Check access rights” use cases in Figure 4.3.

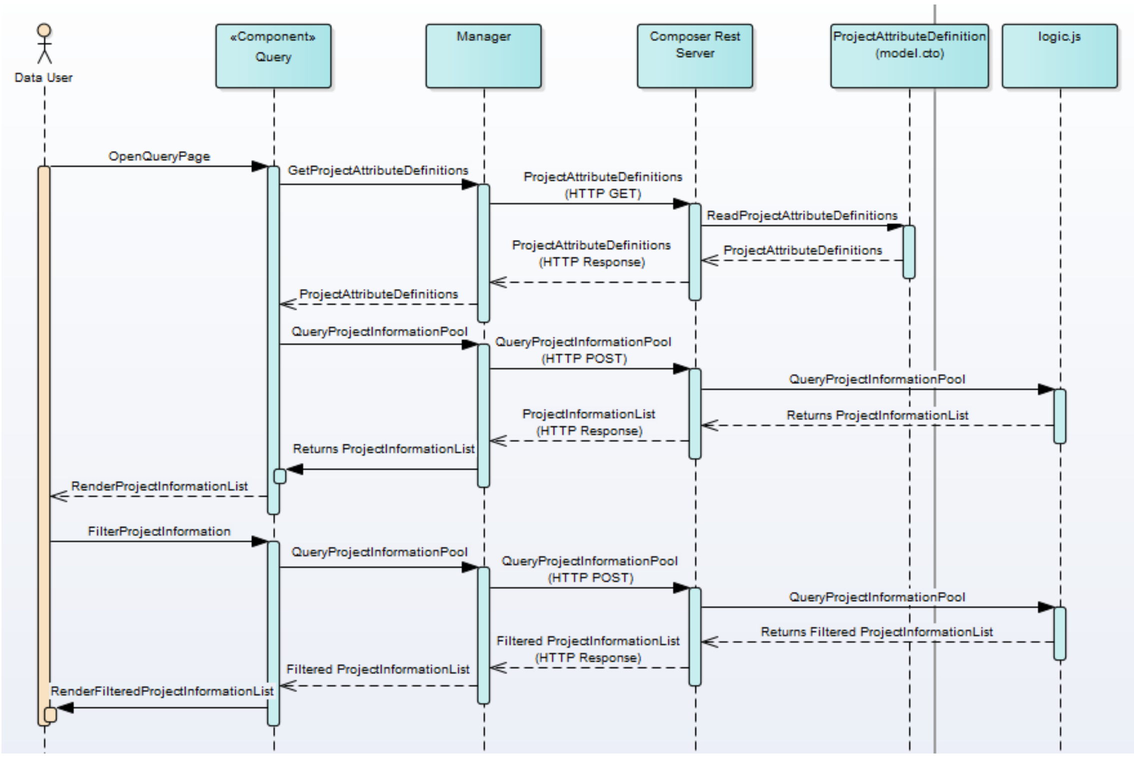


Figure 5.11. Query Project Information Pool Sequence Diagram

5.4.1. Mantel’s Test

In the query page, projects that are filtered and attributes loaded are selected like in Figure 5.11. “Select Column” button is pressed and the columns are selected to include the test. “Mantel’s Test” button is pressed and Mantel’s test page is opened. Dependent columns are selected with “Select Dependent Columns” combobox. “Run Mantel’s Test” button is pressed. Test results are displayed in the grid shown in Figure 5.7. The sequence diagram showing the interactions of this scenario among the system components is shown in Figure 5.12.

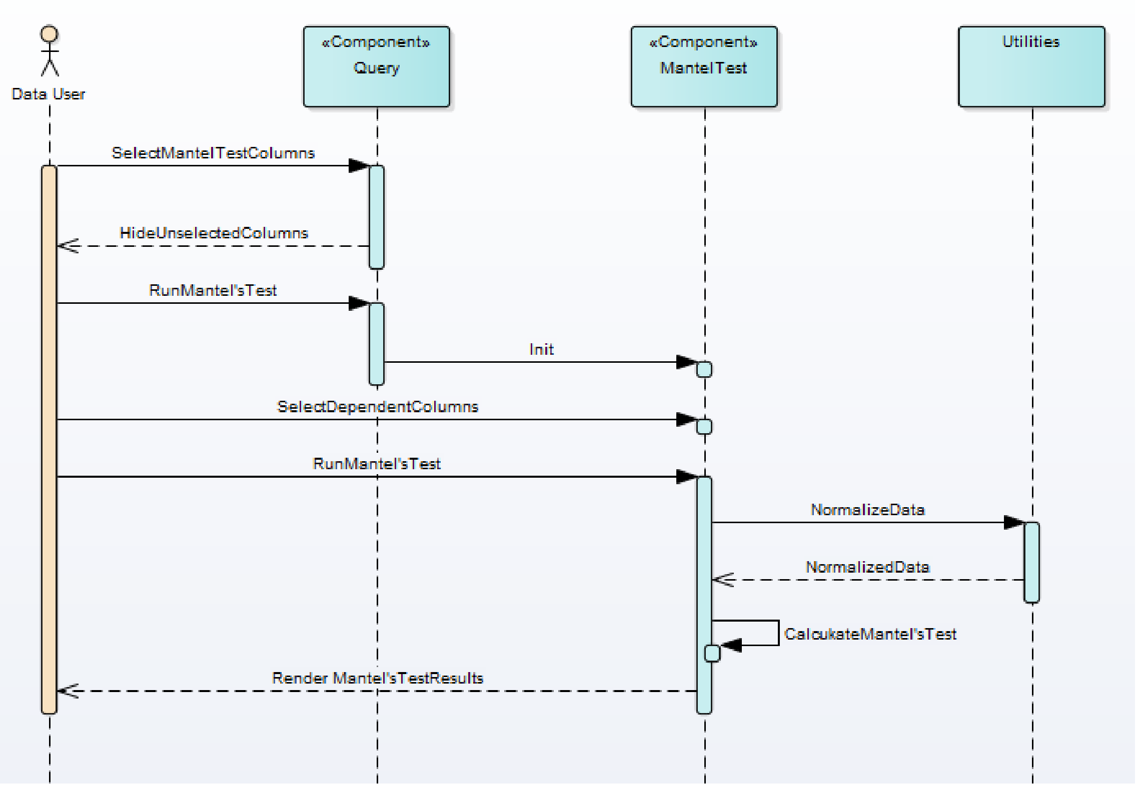


Figure 5.12. Mantel's Test Sequence Diagram

5.4.2. Analogy-Based Estimation

In the query page, projects that are filtered and attributes loaded are selected like in Figure 5.11. “Select Column” button is pressed and the columns are selected to include the estimation. “Estimate By Analogy” button is pressed and analogy-based estimation page is opened. Effort column is selected with “Select Effort Column” combobox. Adjustment column is selected with “Select Adjustment Column” combobox. Input data of the project to be estimated is entered. “Run Analogy-Based Estimation” button is pressed. The result of the estimation is displayed on the grid. If the user enters the actual effort value of the project, statistical results are displayed. The sequence diagram showing the interactions of this scenario among the system components is shown in Figure 5.13.

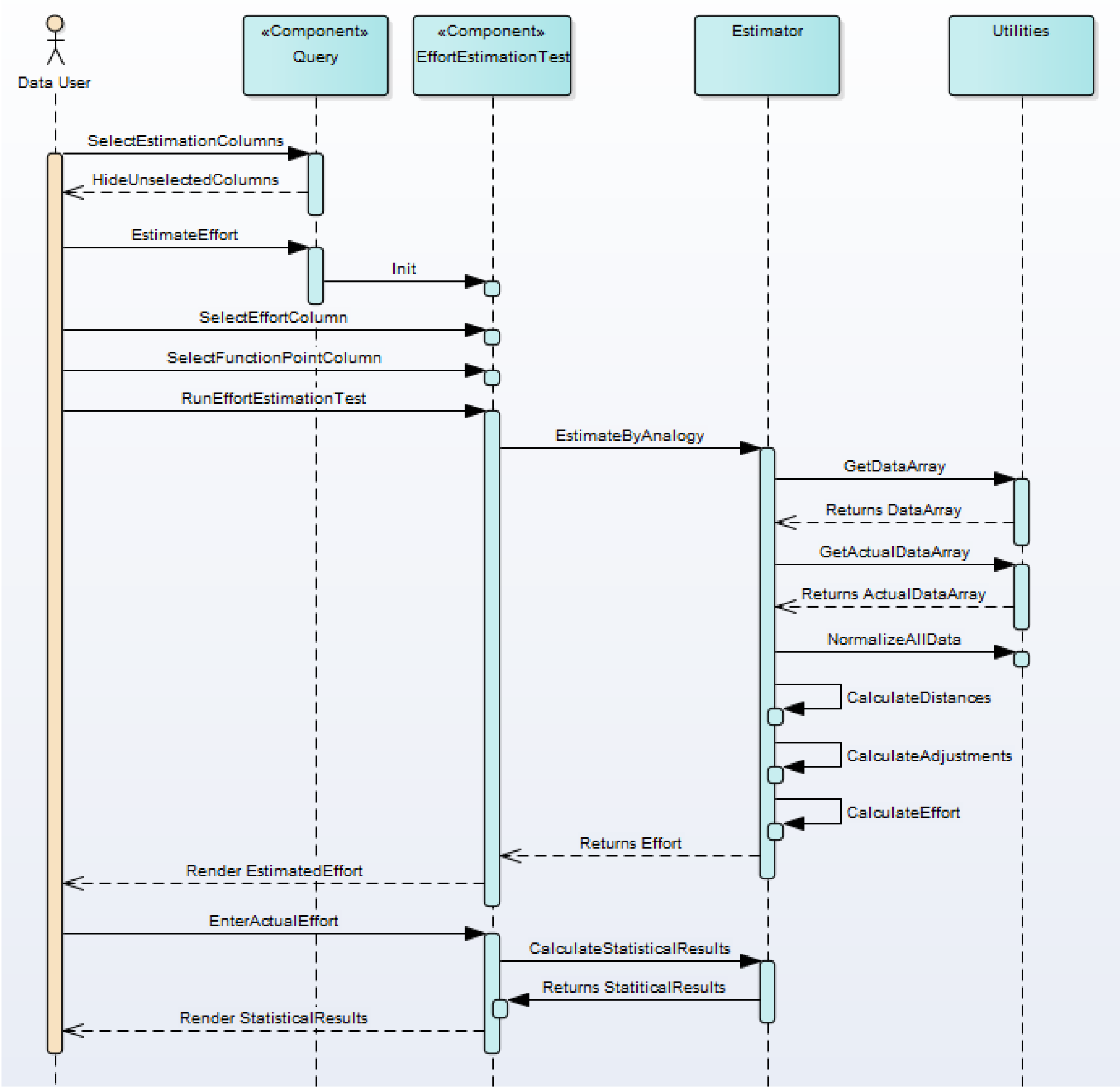


Figure 5.13. Analogy-Based Estimation Sequence Diagram

5.4.3. Estimation Evaluation

In the query page, projects that are filtered and attributes loaded are selected like in Figure 5.11. “Select Column” button is pressed and the columns are selected to include the test. “Evaluate Estimation” button is pressed and estimation evaluation page is opened. Effort column is selected with “Select Effort Column” combobox. Adjustment column is selected with “Select Adjustment Column” combobox. “Evaluate Estimation” button is pressed. Analogy-based effort estimation is performed for each selected projects using other project information. Estimation results and actual effort values are used to calculate statistical results. And estimation and statistical results are displayed in the grid. In addition, the average and median of all estimation results are displayed in another grid.

The sequence diagram showing the interactions of this scenario among the system components is shown in Figure 5.14.

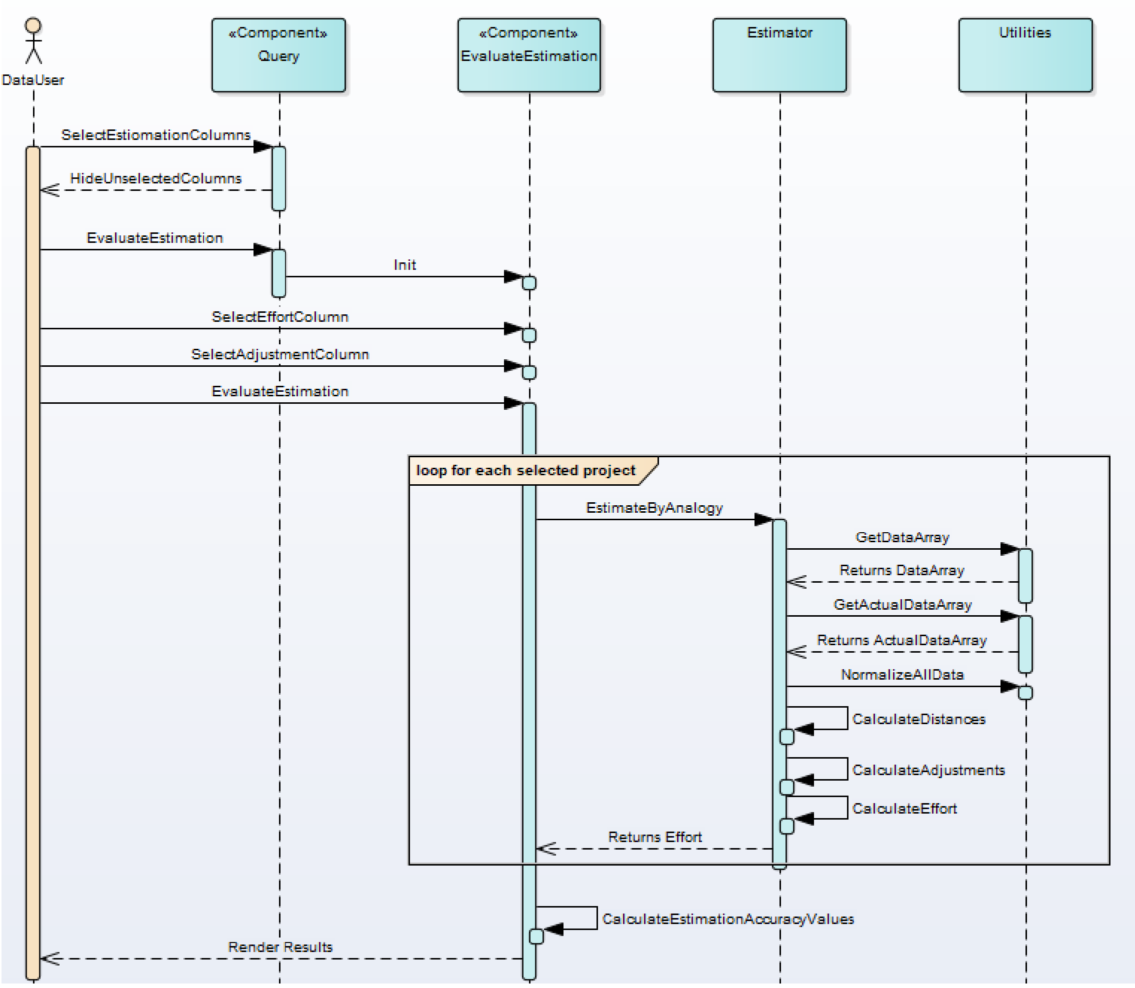


Figure 5.14. Estimation Evaluation Sequence Diagram

6. EVALUATION

In this section, proposed model is evaluated using PoC implementation. In order to evaluate the system, the following research questions are adressed by executing three operational scenarios.

RQ1. Does the system work properly?

RQ2. Does the estimation efficiency improved with the use of the system?

RQ3. Would the implemented system beneficial in terms of estimation performance?

The ISBSG D&E Release April 2016 dataset [1] is used in our evaluations. The projects are filtered to work with high quality project data as in the studies [18][41][42]:

- Projects whose count approach is IFPUG 4+.
- Projects whose normalized effort is equal to summary work effort.
- In these works they take projects which has data quality rating A and B. But in this thesis we only take projects which has data quality rating A for higher quality data.
- Projects with resource level equal to 1.

After taking these projects, our dataset has 307 projects. For evaluation purposes, columns that are given in Table 6.1 are imported.

Table 6.1. Imported Columns from ISBSG Dataset [1]

Column Name	Column Description
“Year of Project”	“Derived from implementation date if known or from other project dates such as project end date, project start date, estimated implementation date, and data compilation date.”
“Industry Sector”	“This is a derived field which attempts to summarise organisation type of the project into a single value.”
“Organisation Type”	“This identifies the type of organisation that submitted the project.”
“Application Type”	“This identifies the type of application being addressed by the project.”
“Development Type”	“This field describes whether the development was a new development, enhancement or re-development.”
“Development Platform”	“Defines the primary development platform.”
“Language Type”	“Defines the language type used for the project.”
“Primary Programming Language”	“The primary language used for the development.”
“Functional Size”	“The unadjusted function point count.”
“Adjusted Function Points”	“The adjusted functional size of the project at the final count.”
“Normalised Work Effort”	“Full life-cycle effort for all teams reported.”
“Speed of Delivery”	“Functional Size Units per elapsed month calculated as: Functional Size / Project Elapsed Time.”
“Project Elapsed Time”	“Total elapsed time for the project in calendar months.”
“Input Count”	“When provided in the submission, this field is the unadjusted functional size of external inputs.”
“Output Count”	“When provided in the submission, this field is the unadjusted functional size of external outputs.”

During the evaluation process; “Functional Size”, “Language Type”, “Development Type”, “Development Platform”, and “Normalised Work Effort” data are generally used as in the study [42].

6.1. Operational Scenarios

Three operational scenarios are defined to evaluate proposed system.

To answer RQ1 and RQ2, Scenario 1 is defined with the activity diagram in Figure 6.1, Figure 6.2, and Figure 6.3. With this scenario, a user imports project data, filters projects and makes analogy-based estimations using the proposed system.

To answer RQ3, Scenario 2 is defined with the activity diagram in Figure 6.4 and Scenario 3 is defined with activity diagram in Figure 6.5. The estimation performances of these two scenarios are compared. While importing project data into the proposed system, a set of rules were defined to make some project attributes inaccessible for the defined user. With these rules, the user can not access all or part of attributes for one third of all projects. In Scenario 2, effort estimation is made by using project data in which all attributes are accessible. In Scenario 3, effort estimation is made by using project data in which all attributes are accessible and the project data to be used in estimation are partially accessible. These two scenarios are intended to demonstrate the benefits of attribute-based access control mechanism provided by the proposed system.

The rating value is used to filter these two kinds of projects easily. For the projects that all attributes are accessible, the rating value is given 90 and above. For the projects in which all or part of the attributes are accessible, the rating value is given below 90. In this way, the two types of project data can be filtered with the rating value.

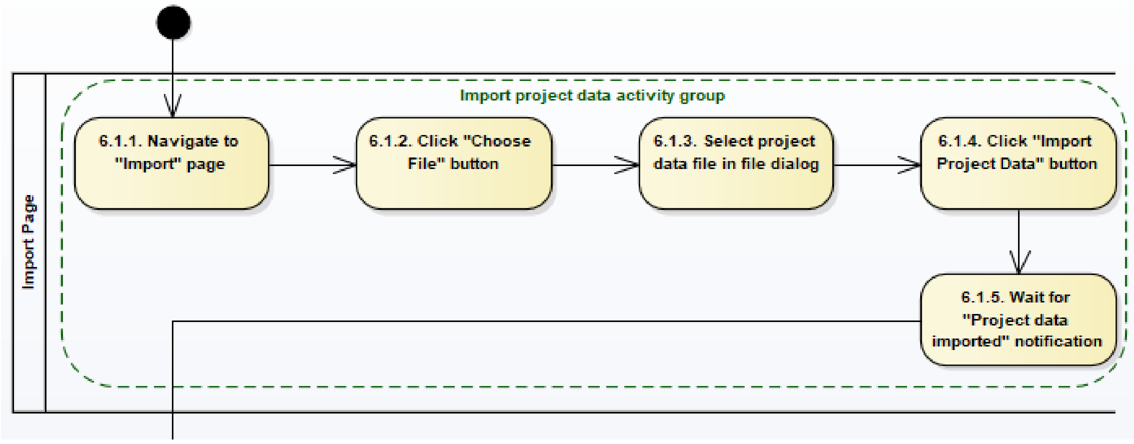


Figure 6.1. Scenario 1 - Activity Diagram: Import Project Data Part

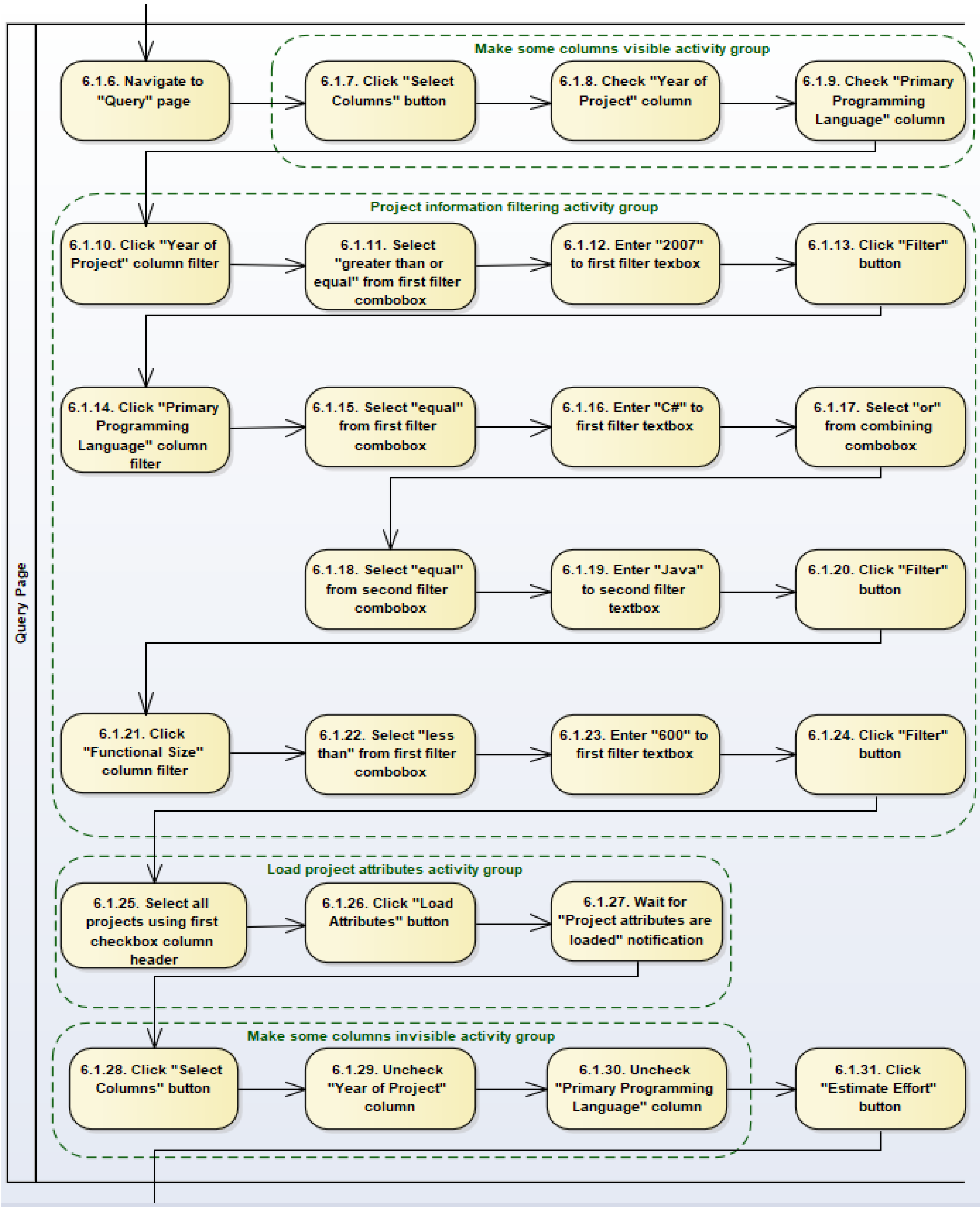


Figure 6.2. Scenario 1 - Activity Diagram: Query Project Information Part

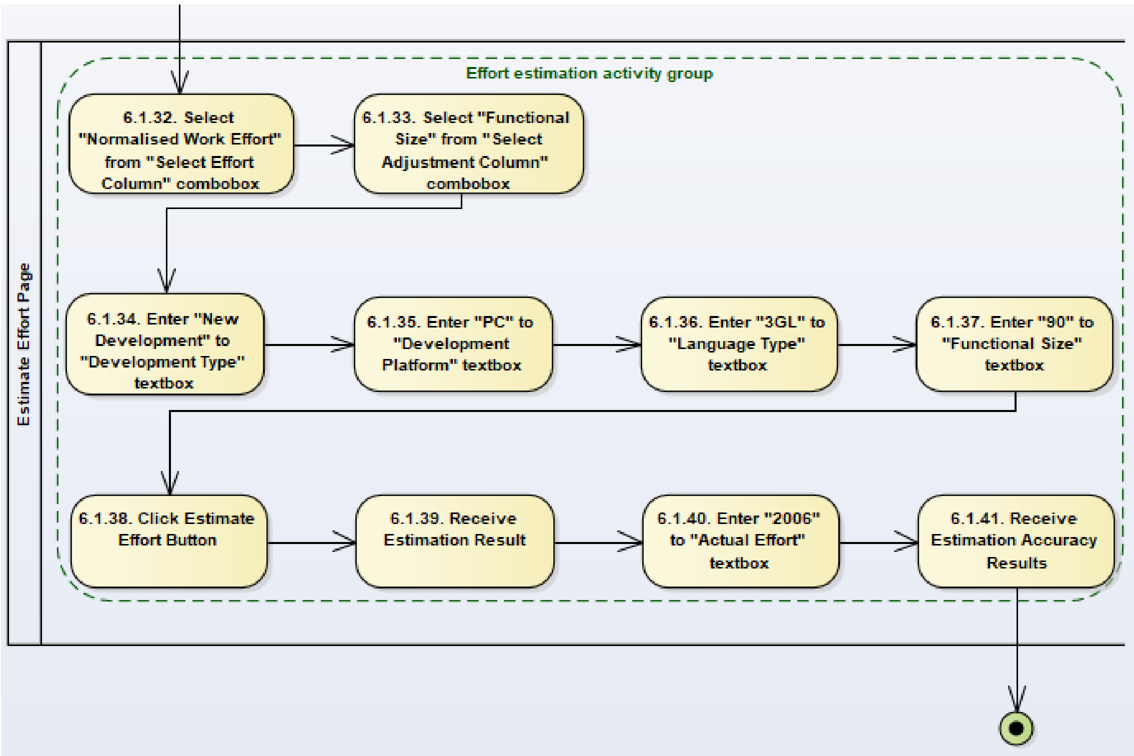


Figure 6.3. Scenario 1 - Activity Diagram: Estimate Effort Part

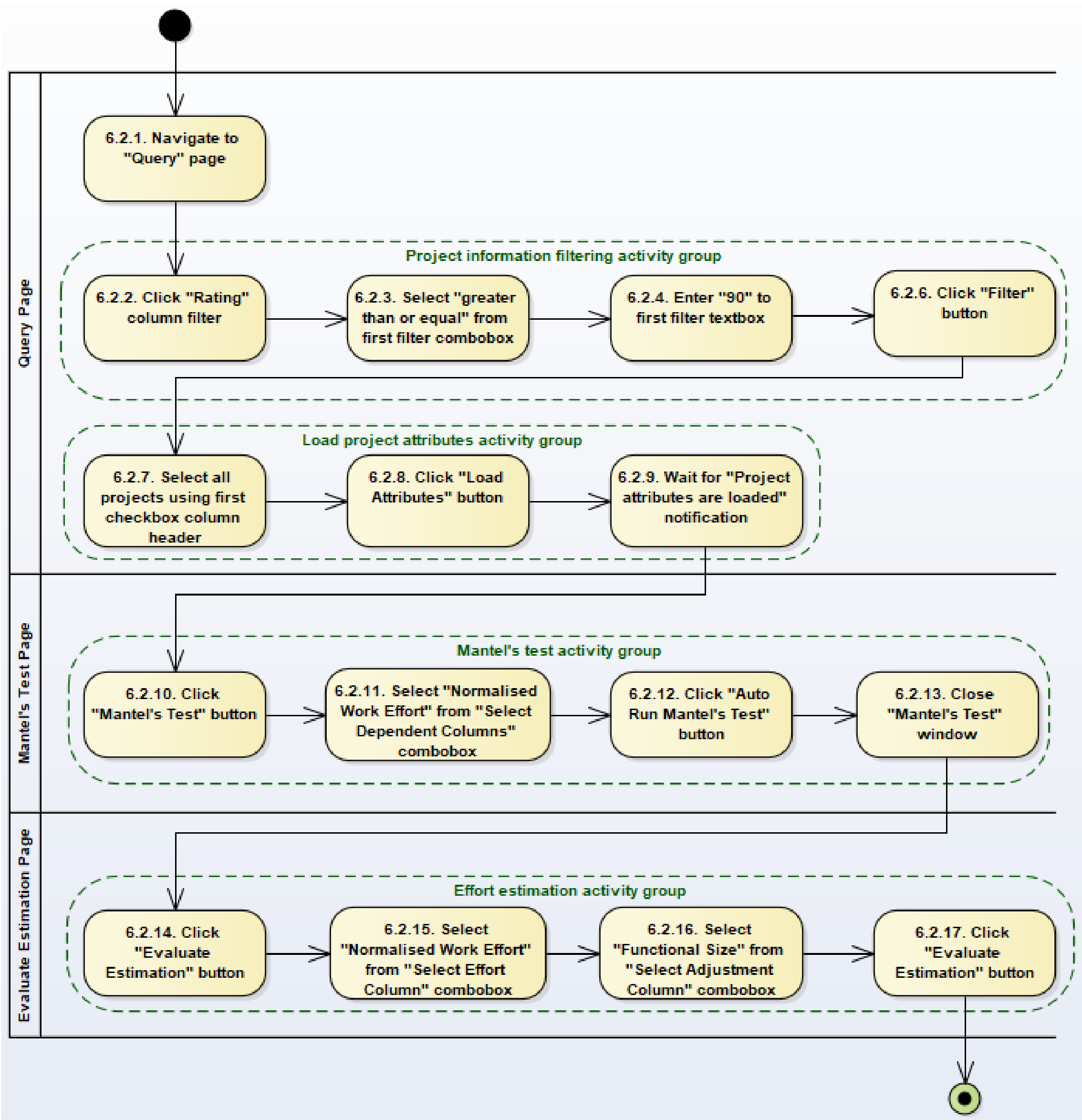


Figure 6.4. Scenario 2 - Activity Diagram: Estimate Effort by Using Fully Accessible Project Data

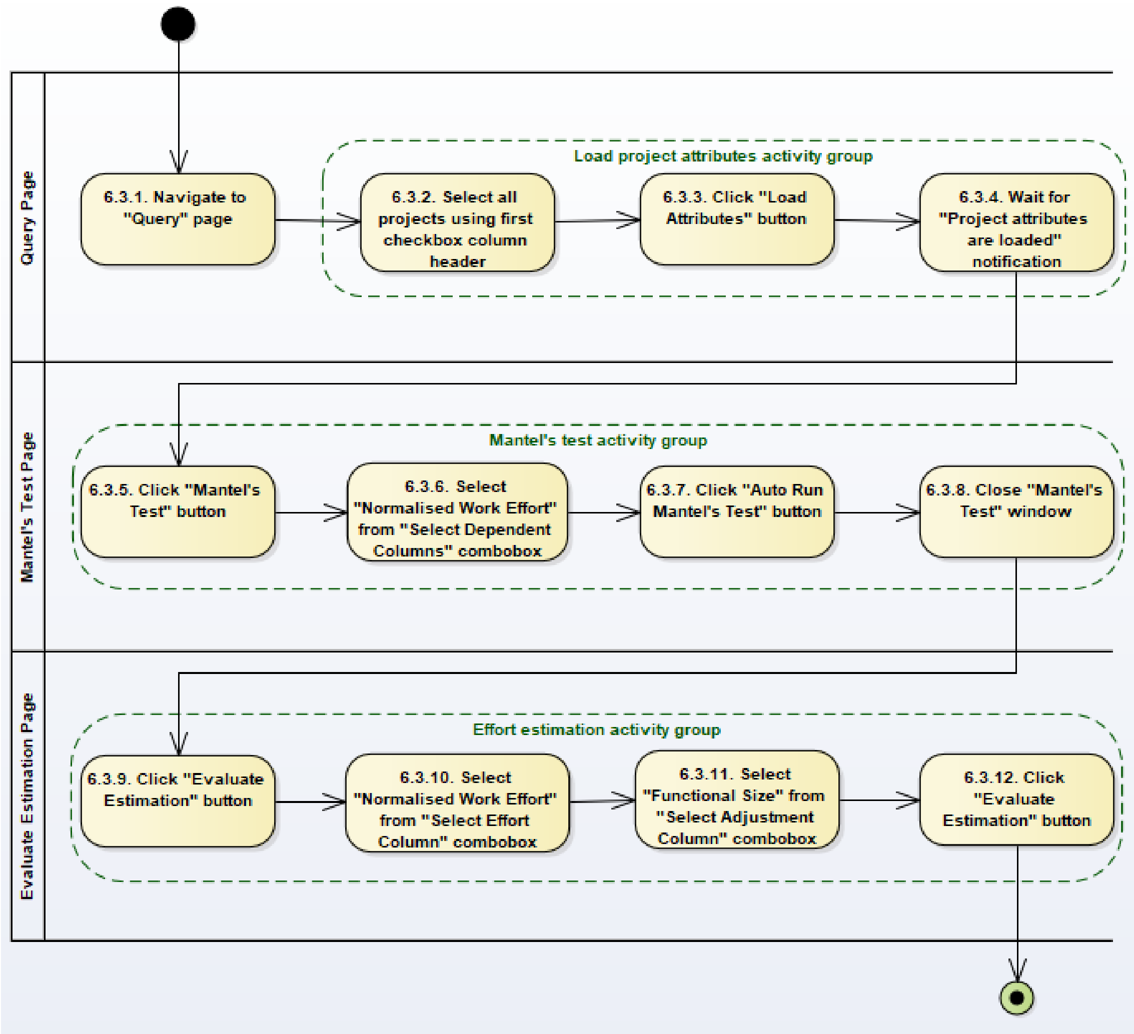


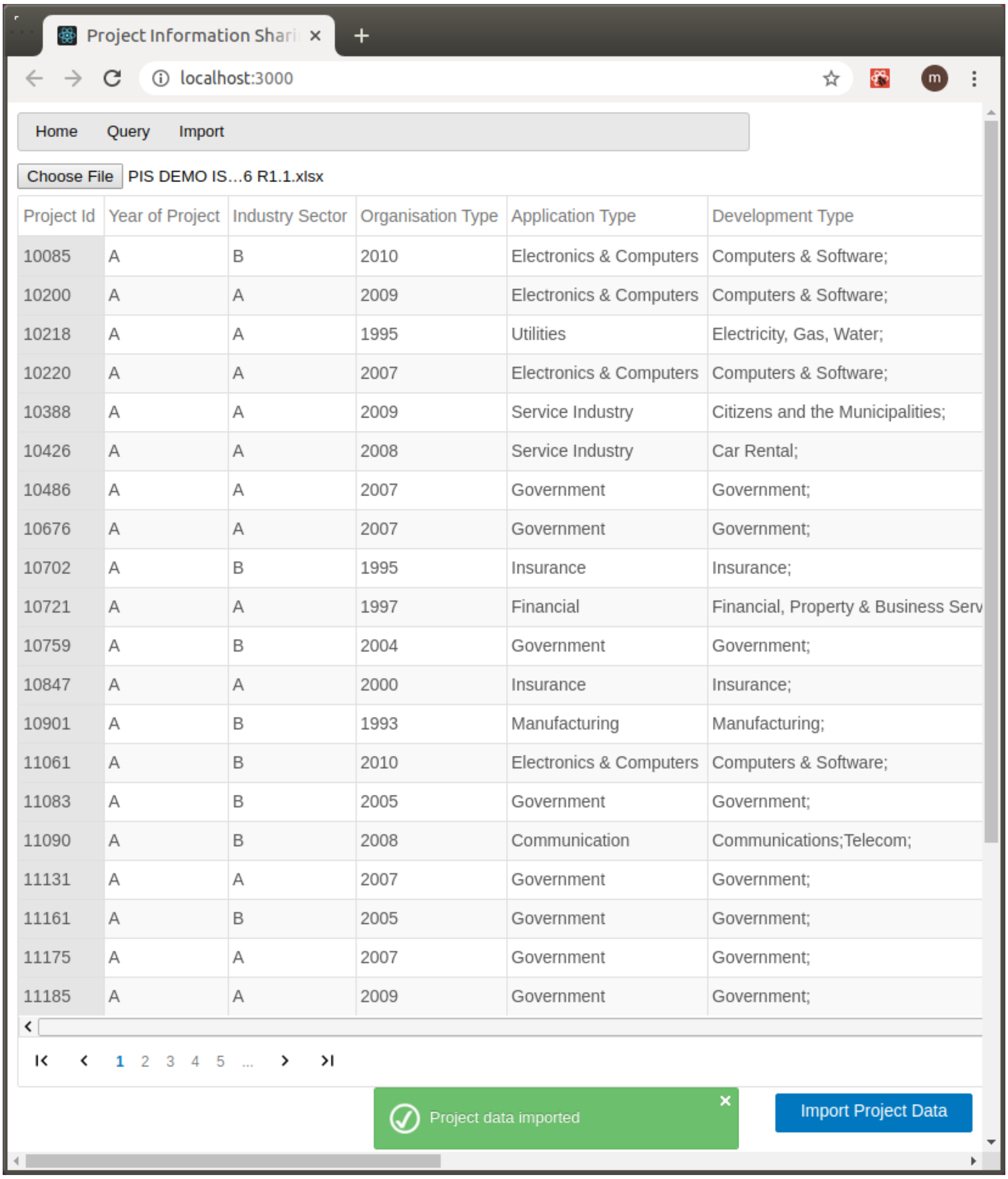
Figure 6.5. Scenario 3 - Activity Diagram: Estimate Effort by Using Fully Accessible and Partially Accessible Project Data

6.2. Execution of Operational Scenarios

6.2.1. Execution of Scenario 1

Scenario 1 is executed by using the activity diagram given in Figure 6.1, Figure 6.2, and Figure 6.3. For this scenario, first, the execution is made using the proposed system for the following activities.

- A user imports project data to the system as in Figure 6.6 as the result of “import project data activity group” (activities 6.1.1 - 6.1.5) given in Figure 6.1.



Project Id	Year of Project	Industry Sector	Organisation Type	Application Type	Development Type
10085	A	B	2010	Electronics & Computers	Computers & Software;
10200	A	A	2009	Electronics & Computers	Computers & Software;
10218	A	A	1995	Utilities	Electricity, Gas, Water;
10220	A	A	2007	Electronics & Computers	Computers & Software;
10388	A	A	2009	Service Industry	Citizens and the Municipalities;
10426	A	A	2008	Service Industry	Car Rental;
10486	A	A	2007	Government	Government;
10676	A	A	2007	Government	Government;
10702	A	B	1995	Insurance	Insurance;
10721	A	A	1997	Financial	Financial, Property & Business Serv
10759	A	B	2004	Government	Government;
10847	A	A	2000	Insurance	Insurance;
10901	A	B	1993	Manufacturing	Manufacturing;
11061	A	B	2010	Electronics & Computers	Computers & Software;
11083	A	B	2005	Government	Government;
11090	A	B	2008	Communication	Communications;Telecom;
11131	A	A	2007	Government	Government;
11161	A	B	2005	Government	Government;
11175	A	A	2007	Government	Government;
11185	A	A	2009	Government	Government;

Figure 6.6. Import Page

- The user navigates to “Query” page and makes visible some columns as in Figure 6.7 as the result of “make some columns visible activity group” (activities 6.1.6 – 6.1.9) given in Figure 6.2.

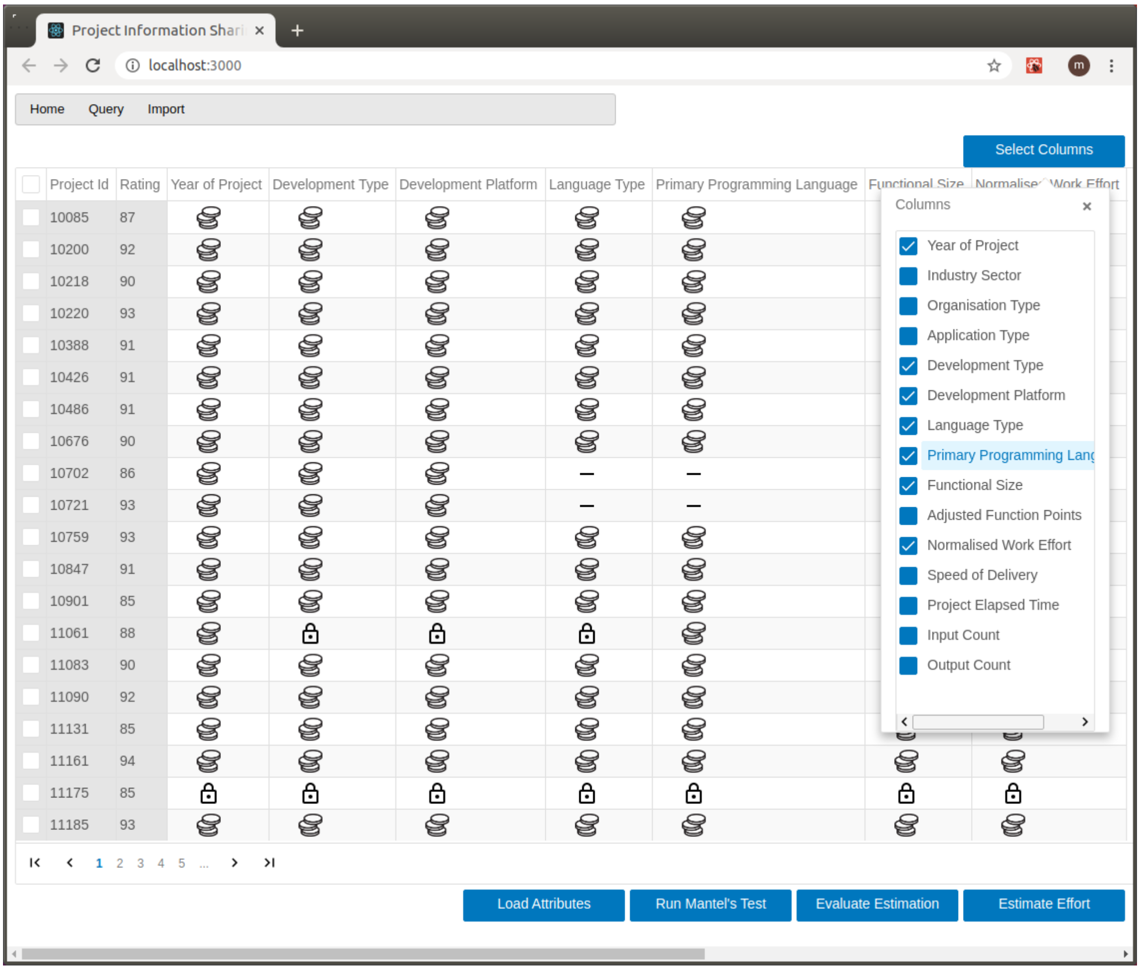


Figure 6.7. Column Selection on Query Page

- The user filters project informations as in Figure 6.8 as the result of "project information filtering activity group" (activities 6.1.10 – 6.1.24) given in Figure 6.2.

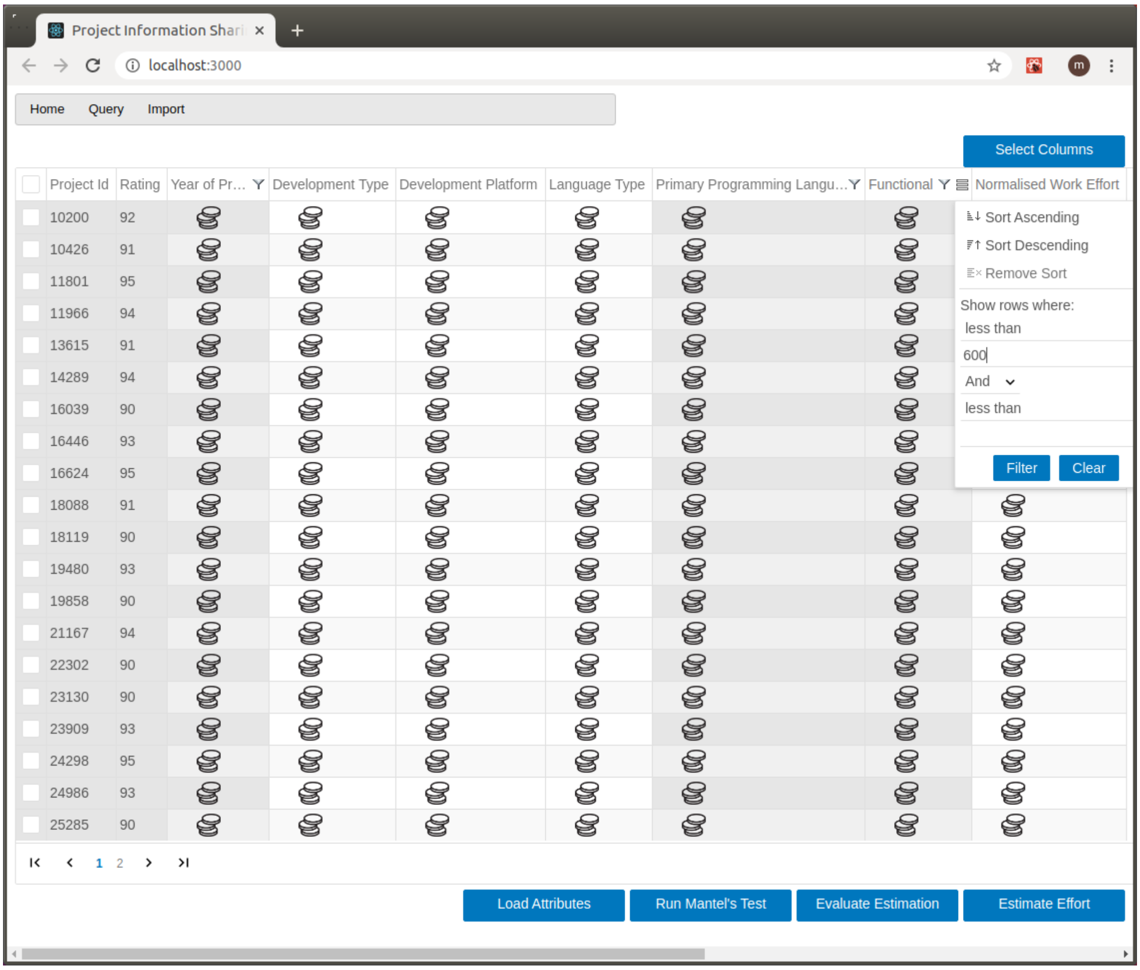


Figure 6.8. Filtering on Query Page

- The user loads project attribute value as in Figure 6.9 as the result of "load project attributes activity group" (activities 6.1.25 – 6.1.27) given in Figure 6.2.

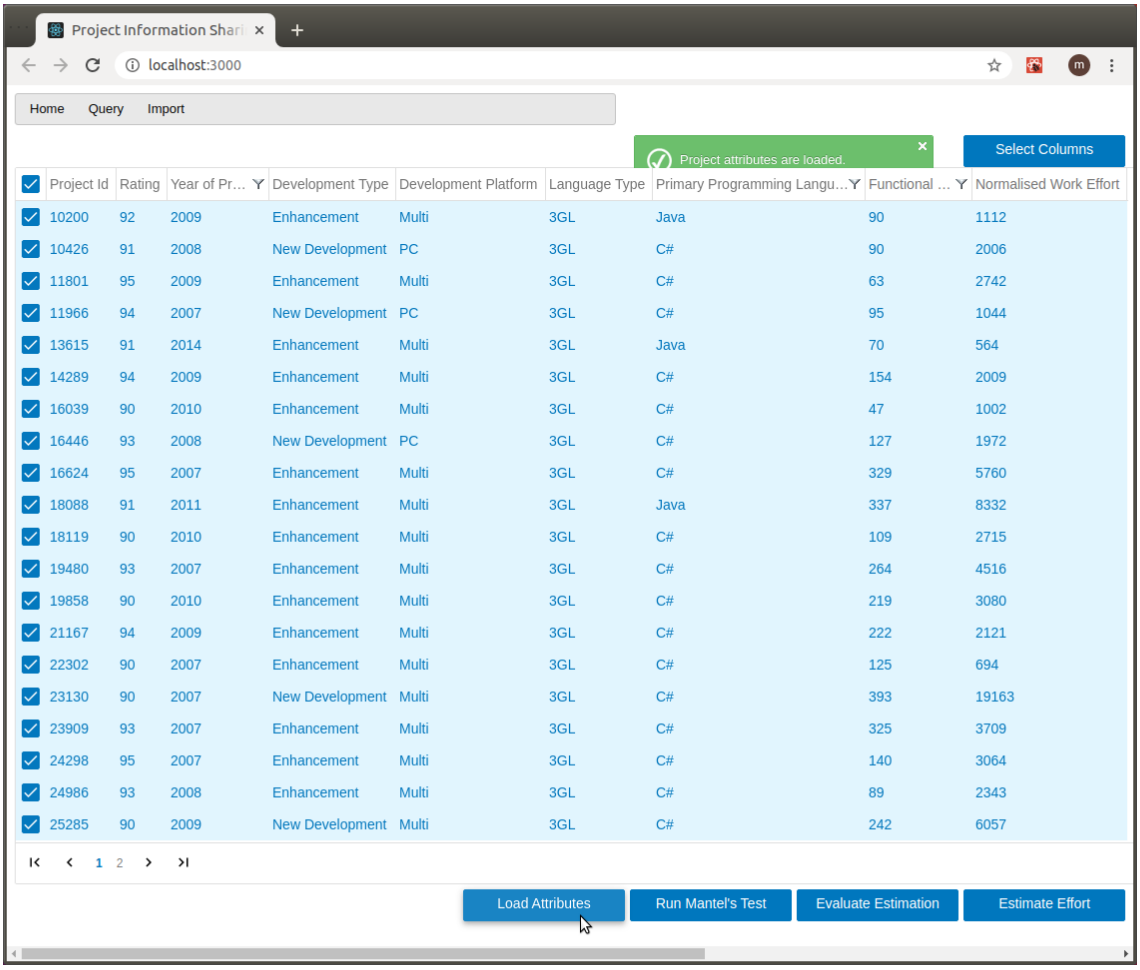


Figure 6.9. Loading Project Attributes on Query Page

- The user makes invisible some columns and opens estimate effort page as in Figure 6.10 as the result of "make some columns invisible activity group" (activities 6.1.28 – 6.1.31) given in Figure 6.2.

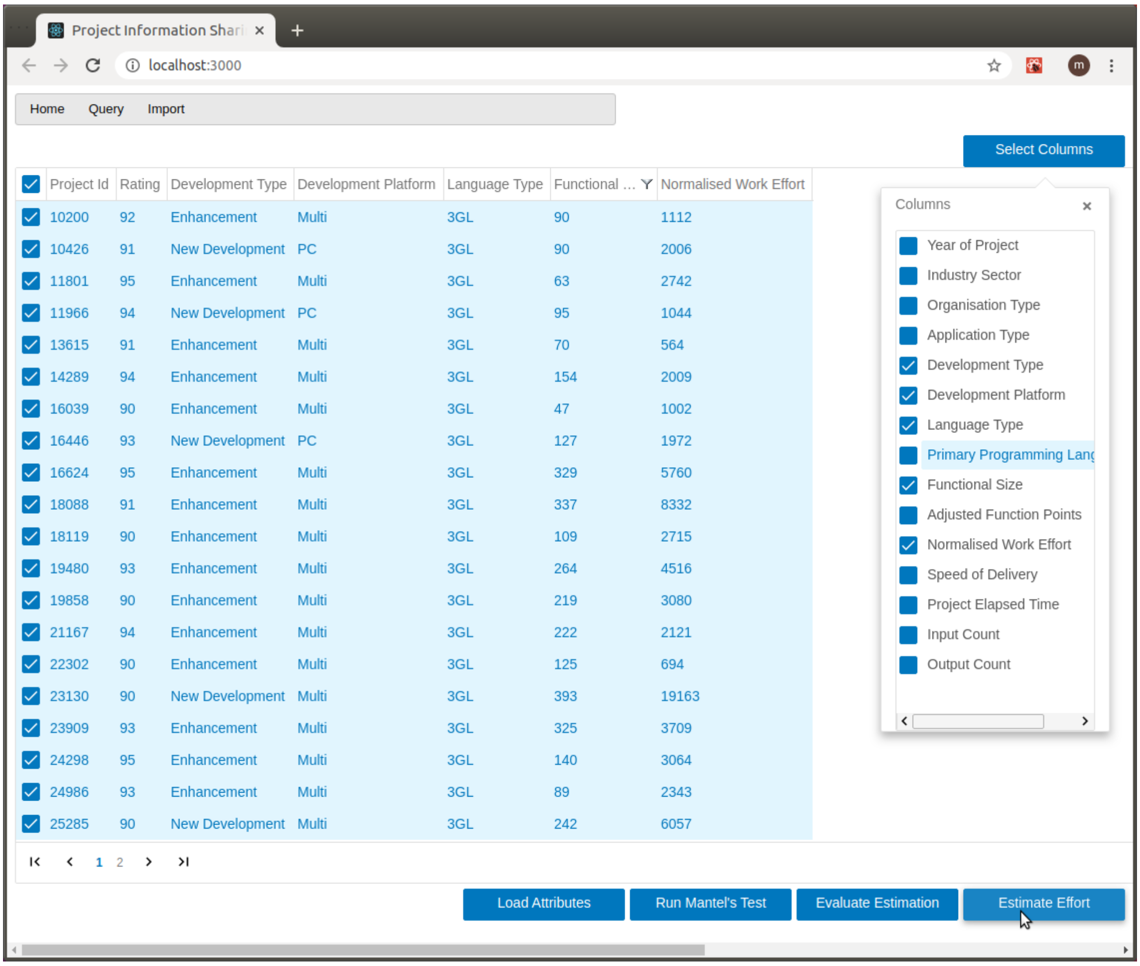


Figure 6.10. Unselecting Columns on Query Page

- The user makes analogy-based estimation as in Figure 6.11 as the result of "effort estimation activity group" (activities 6.1.32 - 6.1.41) given in Figure 6.3.

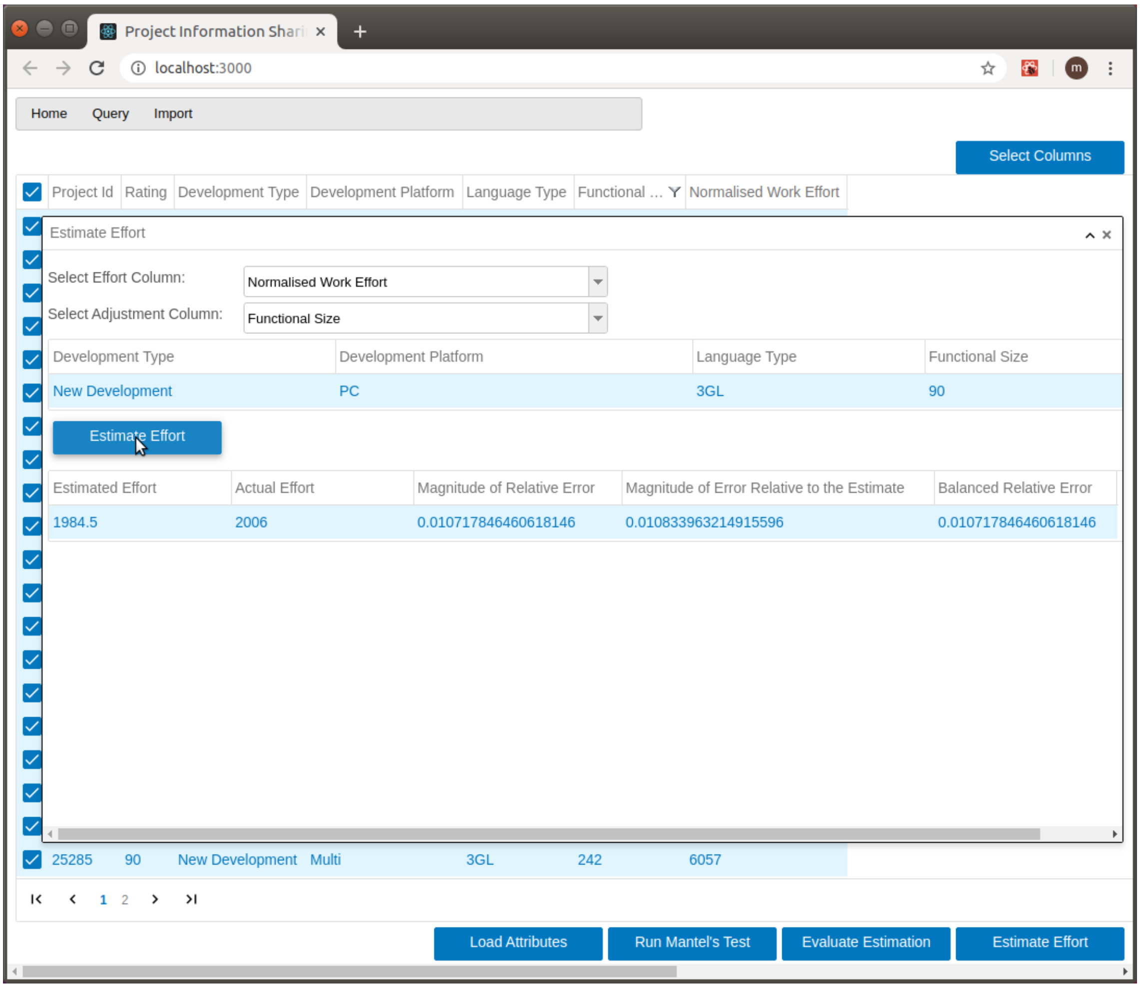


Figure 6.11. Estimate Effort Page

After executing Scenario 1 using our system, the same project filtering and estimation operations are done manually by using the same project data that is kept in MS Excel file by the following steps.

- Projects are filtered as in Table 6.2 with the same filters in "project information filtering activity group" (activities 6.1.10 – 6.1.24) given in Figure 6.2.

Table 6.2. Manually Filtered Projects

Project Id	Development Type	Development Platform	Language Type	Functional Size	Normalised Work Effort
10200	Enhancement	Multi	3GL	90	1112
10426	New Development	PC	3GL	90	2006
11801	Enhancement	Multi	3GL	63	2742
11966	New Development	PC	3GL	95	1044
13615	Enhancement	Multi	3GL	70	564
14289	Enhancement	Multi	3GL	154	2009
16039	Enhancement	Multi	3GL	47	1002
16446	New Development	PC	3GL	127	1972
16624	Enhancement	Multi	3GL	329	5760
18088	Enhancement	Multi	3GL	337	8332
18119	Enhancement	Multi	3GL	109	2715
19480	Enhancement	Multi	3GL	264	4516
19858	Enhancement	Multi	3GL	219	3080
21167	Enhancement	Multi	3GL	222	2121
22302	Enhancement	Multi	3GL	125	694
23130	New Development	Multi	3GL	393	19163
23909	Enhancement	Multi	3GL	325	3709
24298	Enhancement	Multi	3GL	140	3064
24986	Enhancement	Multi	3GL	89	2343
25285	New Development	Multi	3GL	242	6057
25618	New Development	Multi	3GL	298	13489
26156	New Development	Multi	3GL	313	5613
26695	New Development	PC	3GL	24	167
27610	New Development	Multi	4GL	411	10955
28600	Enhancement	Multi	3GL	346	3476
29628	Re-development	Multi	3GL	112	2134
31528	New Development	PC	3GL	90	1963
31628	Enhancement	Multi	3GL	75	976
31683	Enhancement	Multi	3GL	583	1760

- Since other project attributes are categorical, normalization is only performed for function size project attribute as normalized function size column in Table 6.3 by using Equation (2.9).

- The project to be estimate is selected as project with id 10426. The distances with this project are calculated as distance column in Table 6.3 by using Equation (2.10).
- The number of k neighborhood projects to be used during estimation is determined as 2 within the scope of this thesis. Therefore, adjusted effort is calculated for the two closest projects as adjusted effort column in Table 6.3 by using Equation (2.11).
- Finally, effort is estimated with these two adjusted effort values as estimated effort column in Table 6.3 by using Equation (2.12).

Table 6.3. Manual Effort Estimation Values

Project Id	Normalized Function Size	Distance	Adjusted Effort	Estimated Effort
10200	1.18E-01	1		
10426	1.18E-01	0	2006	1984.5
11801	6.98E-02	1.415		
11966	1.27E-01	0.008		
13615	8.23E-02	1.414		
14289	2.33E-01	1.418		
16039	4.11E-02	1.416		
16446	1.84E-01	0.066		
16624	5.46E-01	1.477		
18088	5.60E-01	1.481		
18119	1.52E-01	1.414		
19480	4.29E-01	1.448		
19858	3.49E-01	1.432		
21167	3.54E-01	1.433		
22302	1.81E-01	1.415		
23130	6.60E-01	1.137		
23909	5.38E-01	1.475		
24298	2.08E-01	1.417		
24986	1.16E-01	1.414		
25285	3.90E-01	1.036		
25618	4.90E-01	1.066		
26156	5.17E-01	1.076		
26695	0.00E+00	0.118		
27610	6.92E-01	1.153		
28600	5.76E-01	1.486		
29628	1.57E-01	1.414		
31528	1.18E-01	0	1963	
31628	9.12E-02	1.414		
31683	1.00E+00	1.666		

6.2.2. Execution of Scenario 2

Scenario 2 is executed by using the activity diagram given in Figure 6.4. For this scenario, the execution is made using the proposed system for the following activities.

- Projects in which all attributes are accessible are filtered with the rating column as in Figure 6.12 as the result of "project information filtering activity group" (activities 6.2.1 – 6.2.6) given in Figure 6.4

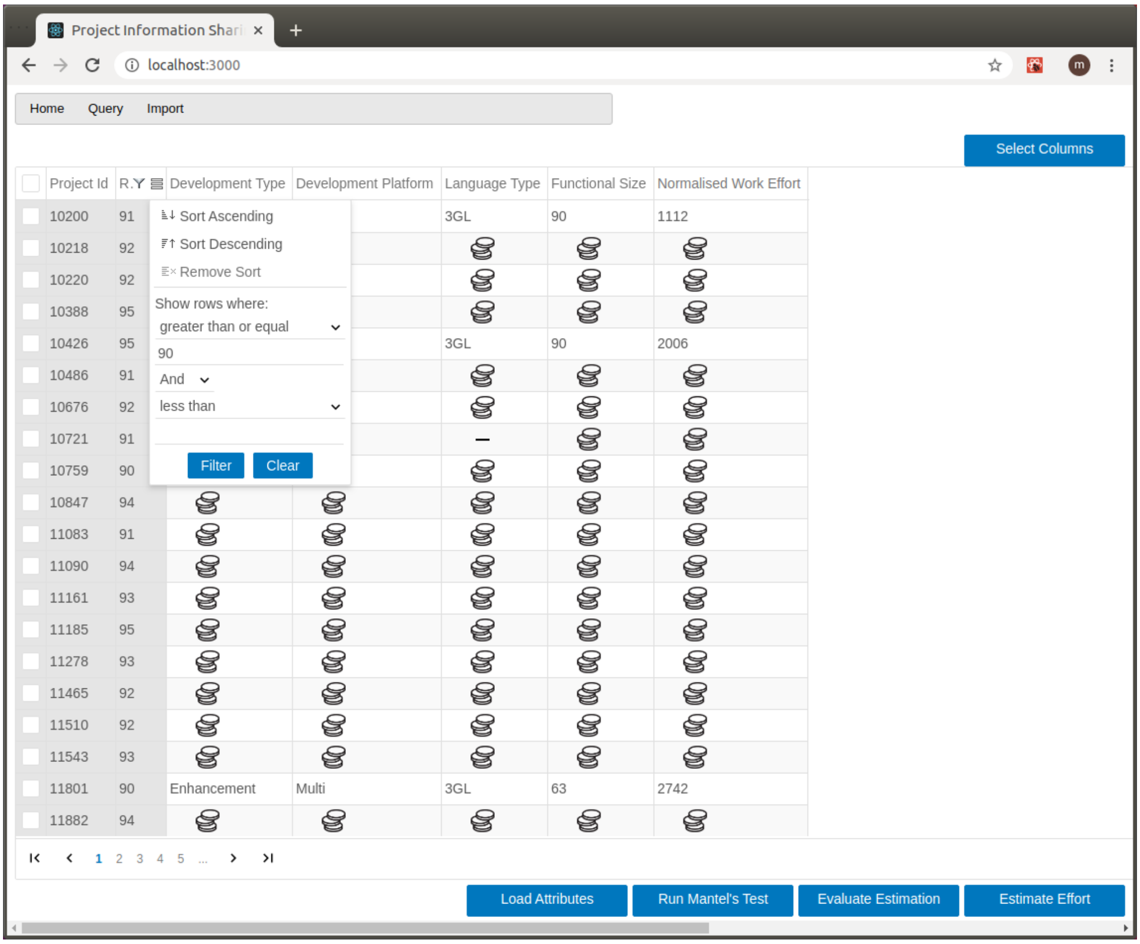


Figure 6.12. Filtering on Query Page

- All projects as a result of the rating filter are selected and loaded as in Figure 6.13 as the result of "load project attributes activity group" (activities 6.2.7 – 6.2.9) given in Figure 6.4.

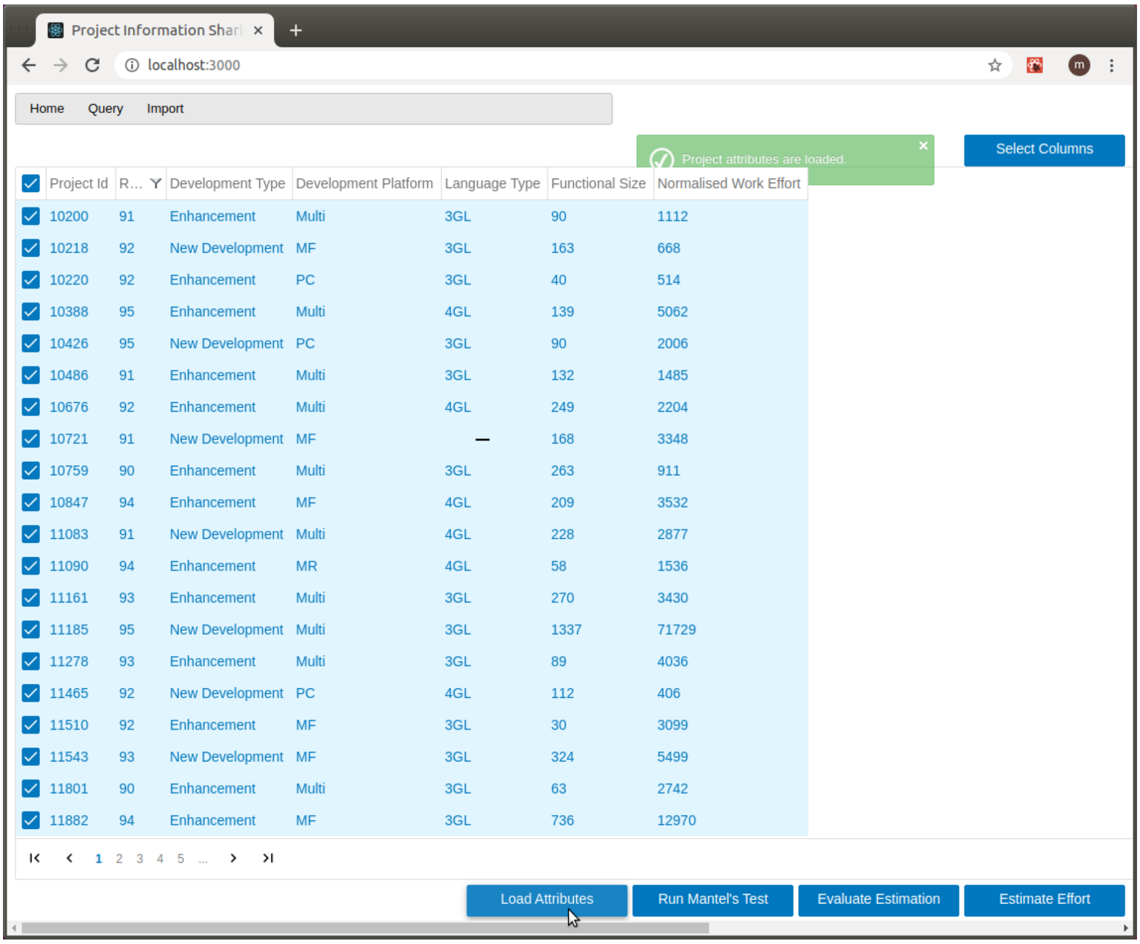


Figure 6.13. Loading Project Attributes on Query Page

- Mantel’s test is performed to filter outlier projects as in Figure 6.14 as the result of “Mantel’s test activity group” (activities 6.2.10 – 6.2.13) given in Figure 6.4.

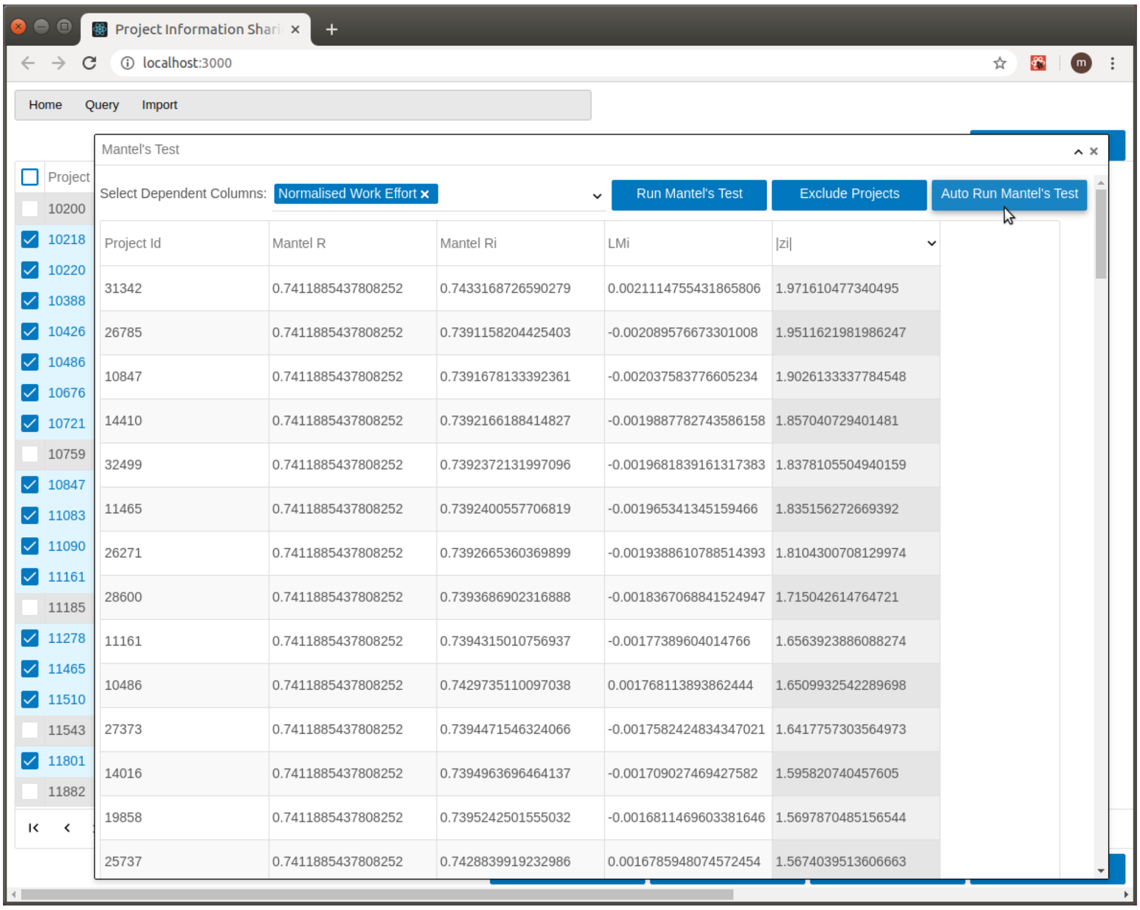


Figure 6.14. Mantel’s Test Page

- Analogy-based estimation is performed for all remaining projects after outlier filtering. For each project, effort estimation is made with excluding current project from estimation project dataset. Then accuracy results are calculated as in Figure 6.15 as the result of “effort estimation activity group” (activities 6.2.14 – 6.2.17) given in Figure 6.4.

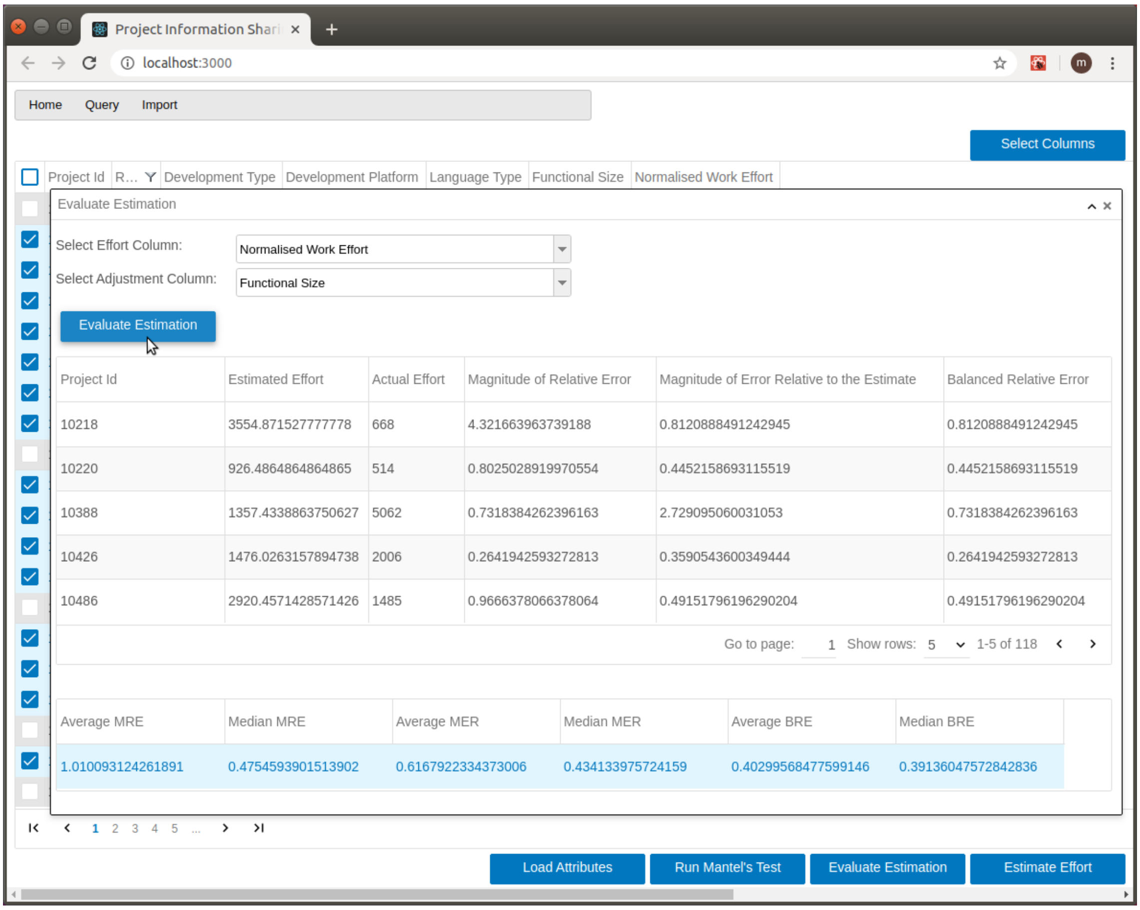


Figure 6.15. Evaluate Estimation Page

6.2.3. Execution of Scenario 3

Scenario 3 is executed by using the activity diagram given in Figure 6.5. For this scenario, the execution is made with the proposed system for the following activities.

- All column filters are cleared and all projects are selected and loaded as in Figure 6.16 as the result of “load project attributes activity group” (activities 6.3.1 – 6.3.4) given in Figure 6.5 .

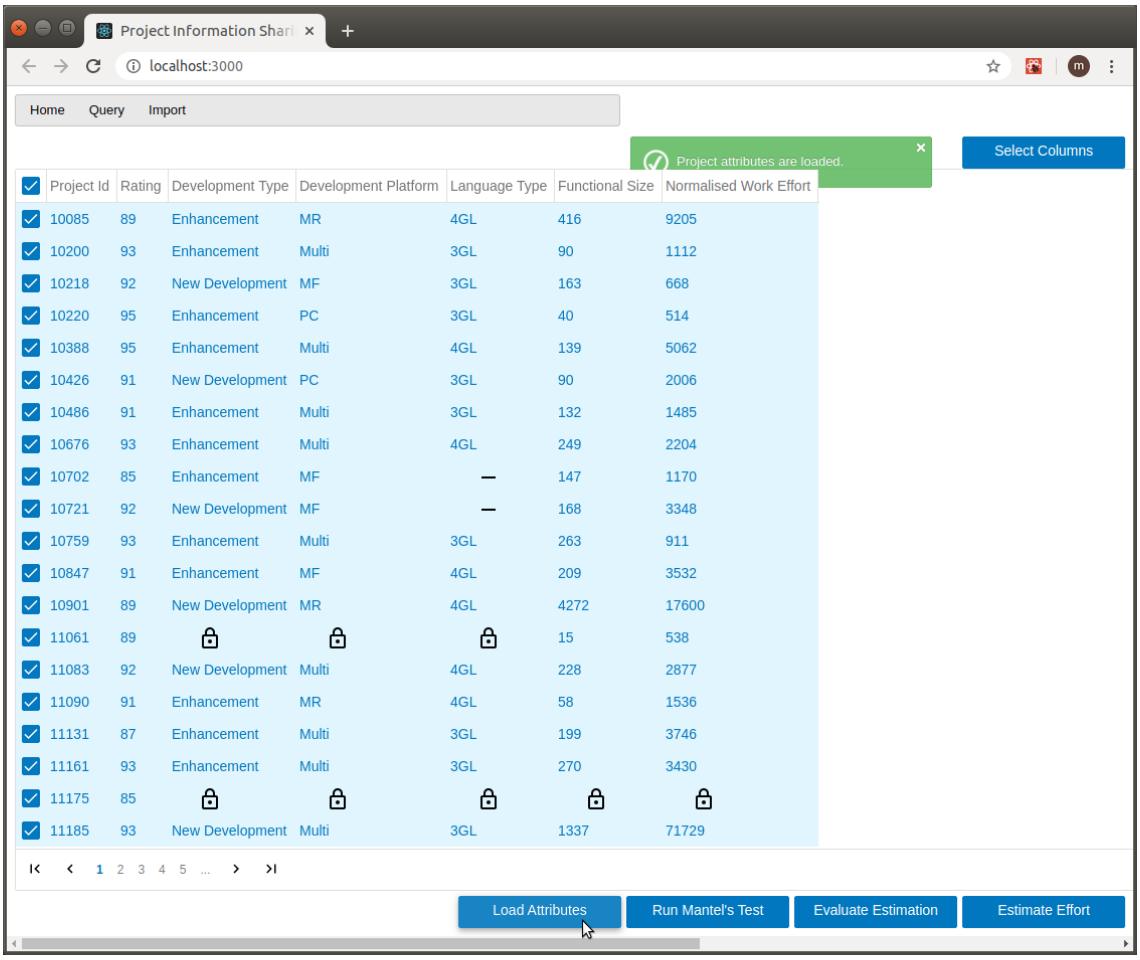


Figure 6.16. Loading Projects on Query Page

- Mantel’s test is performed to filter outlier projects as in Figure 6.17 as the result of “Mantel’s test activity group” (activities 6.3.5 – 6.3.8) given in Figure 6.5.

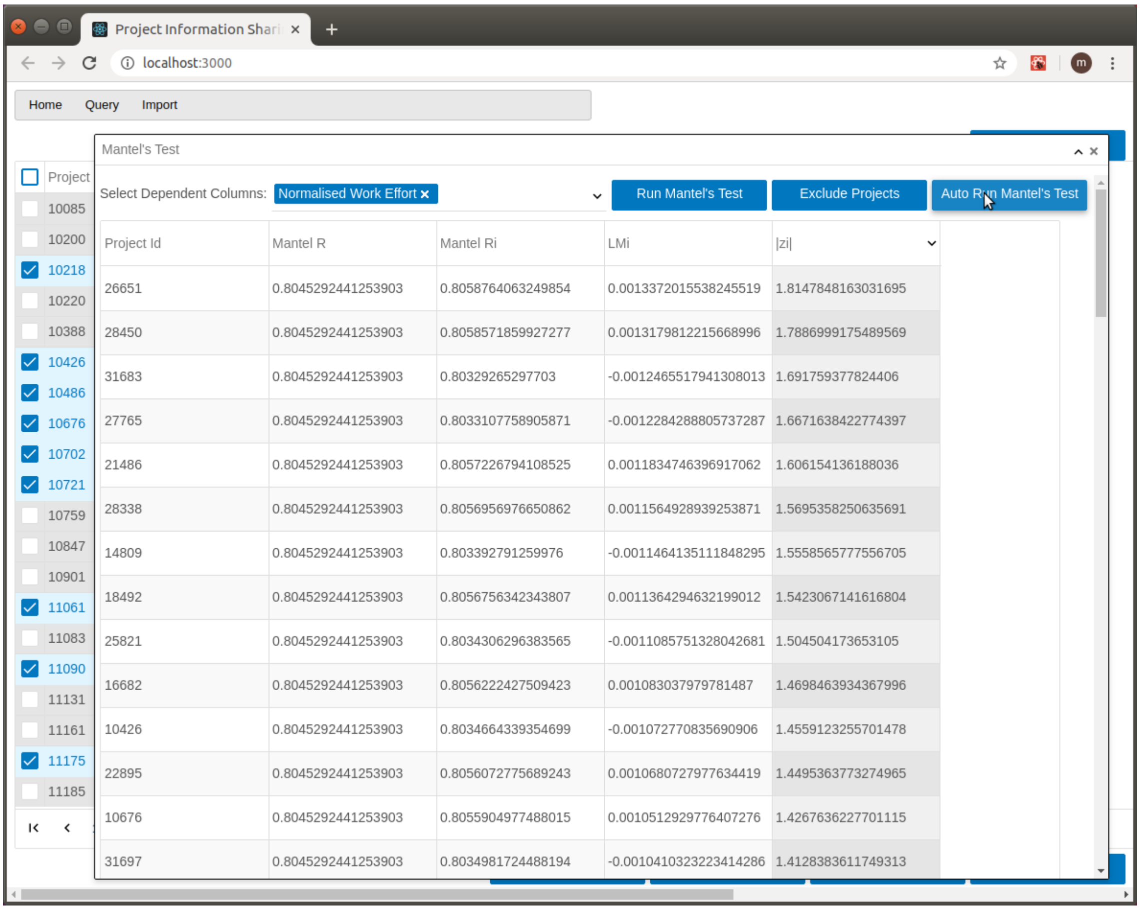


Figure 6.17. Mantel’s Test Page

- Analogy-based estimation is performed for all remaining projects after outlier filtering. For each project, effort estimation is made with excluding current project from estimation project dataset. Then accuracy results are calculated as in Figure 6.18 as the result of “effort estimation activity group” (activities 6.3.9 – 6.3.12) given in Figure 6.5.

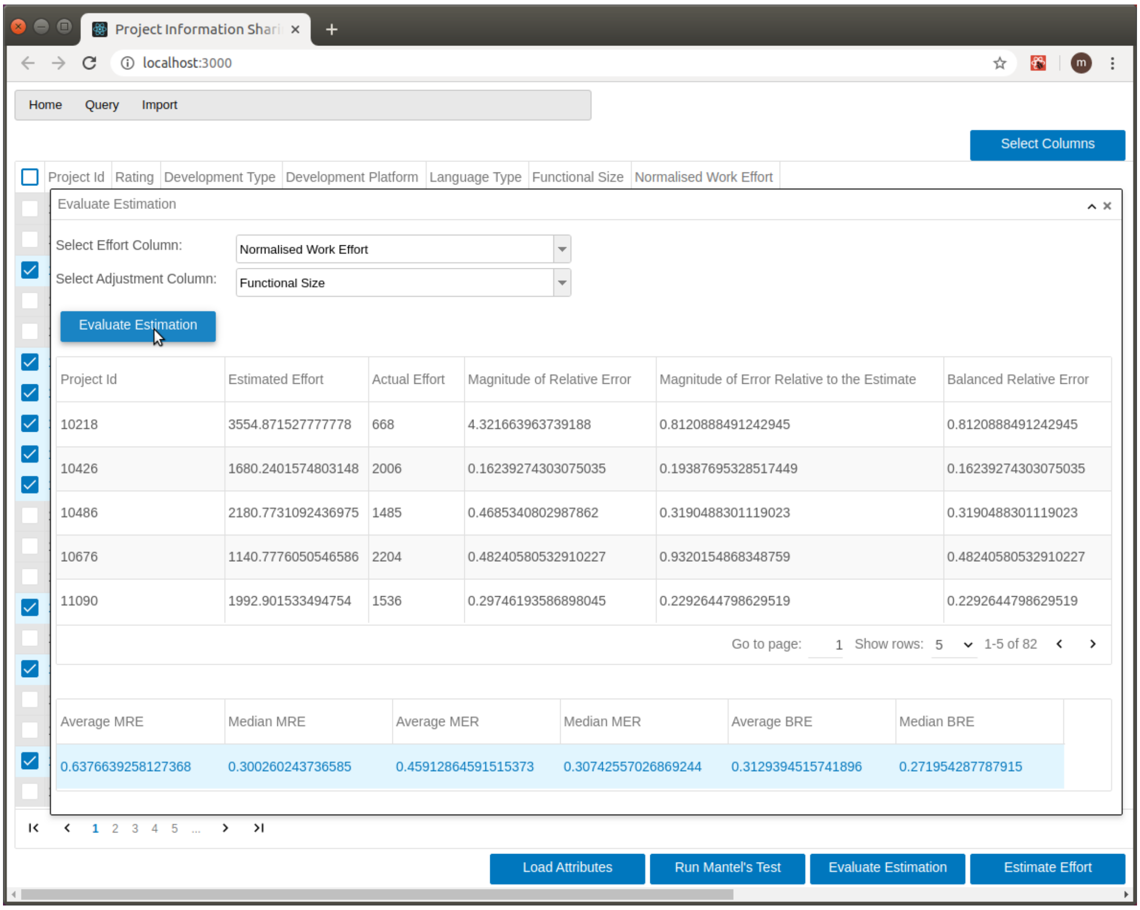


Figure 6.18. Evaluate Estimation Page

6.3. Evaluation

Answer to RQ1. Does the system work properly?

After execution of Scenario 1, it was seen that the estimation result made with proposed system (in Figure 6.11) and the estimation result made mannually (in Table 6.3) were equal (1984.5). For RQ1, we can say that the proposed system works correctly.

Asnwer to RQ2. Does the estimation efficiency improved with the use of the system?

The user, who executed the Scenario 1 using the proposed system, used the system for the first time. The execution duration was measured as 3 minutes and 56 seconds.

On the other hand, the durations recorded for manual execution steps are listed below. The total execution duration for 29 projects was recorded as approximately 55 minutes.

- Filtering for 3 project attributes for 29 projects: approximately 7 minutes.
- Normalization calculations for 1 project attribute: approximately 8 minutes.
- Distance calculations for 1 normalized project attribute: approximately 30 minutes.
- Effort Calculation: 10 seconds.

These execution duration results provide an answer to RQ2. Accordingly, the proposed system can be said to increase time-efficiency in filtering and effort estimation processes.

Answer to RQ3. Would the implemented system beneficial in terms of estimation performance?

Scenario 2 illustrates the use of existing datasets. Since these datasets do not have an access control mechanism, all users authorized to these datasets can access all project data. Data providers, who want to keep some parts of the project data private and make other parts accessible to all or specific users, do not add project data to these datasets. This leads to less (208 in Scenario 2) project data being collected.

Scenario 3 illustrates one of the beneficial uses of the proposed system. The proposed model allows the data provider to make some parts of the project data accessible to all or specific users. In this way, according to the access permissions of the data users, it is possible to access more project data (257 versus 208) than in Scenario 2. Since this will increase the number of project attributes used during estimation, it is expected to improve estimation performance.

The estimation accuracy results of Scenario 2 and Scenario 3 in Table 6.4 show that the proposed system in response to RQ3 is beneficial in terms of estimation performace. All estimation performance values of Scenario 3 are better than Scenario 2.

Table 6.4. Execution Results of Scenario 2 and Scenario 3

Scenario	**Number of Projects**	**Average MRE**	**Median MRE**	**Average MER**	**Median MER**	**Average BRE**	**Median BRE**
Scenario 2	208	1.01	0.47	0.61	0.43	0.40	0.39
Scenario 3	257	0.63	0.30	0.45	0.30	0.31	0.27

With the proposed system, it is seen that more project data can be collected and thus, the posibility of accessing more suitable project data for accurate estimation increases.

Outlier project data, which may increase as project data increases, may adversely affect the estimation performance. But this effect can be minimized by using outlier detection algorithms such as Mantel's test sampled in Scenario 2 and Scenario 3.

7. CONCLUSION

7.1. Overview

In this thesis, we proposed a conceptual model for blockchain-based software project information sharing to encourage stakeholders for sharing and using project information by defining an access control mechanism. With this mechanism, only data owner can manage the access controls of the project data. In order to make stored project information more reliable, an incentive mechanism that benefits all roles is employed. The features of blockchain technology make the model more secure and reliable.

We implemented a proof-of-concept full-stack application using blockchain to evaluate proposed model. For this purpose, the following three research questions we prepared and the operational scenarios to answer these questions were defined.

RQ1. Does the system work properly?

RQ2. Does the estimation efficiency improved with the use of the system?

RQ3. Would the implemented system be beneficial in terms of estimation performance?

We answered RQ1 by comparing the estimation results made with the proposed system and with manual estimation. The fact that the estimation results made by these two methods are equal shows that our system is working properly.

To answer RQ2, we measured durations of the autmated and manual estimation methods. The measured execution duration for the proposed system is 3 minutes 56 seconds and for the manual method is approximately 55 minutes. These measured durations show that our system increases time-efficiency.

To answer RQ3, we made two effort estimations with different project data. First, we made an effort estimation by using fully accessible project data. Then, we made another effort estimation by using fully accessible and partially accessible project data. The second estimation results are found to be better than the previous estimation results. Since our system allows to make some parts of the project data accessible, this made it possible to access more project data (257 versus 208) in the second estimation. These estimation results show that our system is beneficial in terms of estimation performance.

7.2. Limitations

During the PoC implementation, limitations caused by Hyperledger Composer were encountered. Since the API provided for querying the stored project information is very limited, Hyperledger Fabric native API access was made via Hyperledger Composer for some queries such as starts with, contains. This kind of access ignores access control rules defined in .acl file of Hyperledger Composer, and therefore access control rules was implemented manually. In addition, no method was found for pagination of project data with Hyperledger Composer.

On the side of the problem domain, there is a need for a standard data model for software project estimation which is a tactical level issue. As a future work, the data owned can be processed with respect to this data model by smart contracts.

7.3. Threats to System Realization and Suggestions

Threats and suggestions that can be encountered during a real system development using the proposed PoC system are described below.

Threat-1: Users in all roles may want to damage the proposed system. A malicious data provider may add incorrect project information to the system. The rating values of the projects added maliciously are determined low by verifiers, so that the wrong project data can be filtered using the rating value. In order to determine the project rating value correctly, it will be appropriate to select verifiers from reliable data providers with similar project data. A malicious data user may make a denial of service (DoS) attack through prject filtering to decrease system availability. Although this type of attack can detect from transaction logs, it would be appropriate to select data users from data providers to prevent the attack.

Threat-2: Users with privacy concerns may refrain from participating the proposed system. Convincing these users that the system is secure is one of the most difficult problems to increase the likelihood of using the system. To address this problem, stakeholders can be informed about the security of the system. In addition, evidence of the ineffectiveness of attacks on the system can be provided to users.

Threat-3: Blockchain is still a developing technology like Hyperledger Fabric, which is used in the proposed system application. Due to the ongoing development, releases used within the system may contain bugs. For this reason, the relases of the blockchain technology used for the application should be followed, new features should be used and necessary updates should be made to fix the bugs. While aplying newly developed features that will benefit the system, introduced bugs should be resolved.

Threat-4: After project information sharing business definition is deployed to the Hyperledger Fabric network, access rules must be defined so that network admin and channel admin cannot access project data. Due to the nature of the blockchain, only the users allowed by the data owner should be able to access the project data, and no system authority should access it.

Threat-5: The PoC system is used in a way that a specified user can access. In a real scenario the Composer Rest Server should be generated in a way suitable for multiple users. A user creates a membership request to access the system. The system creates a participant entity for the user who makes the membership request and issues an identity. The user can add the identity to the wallet via the client application to access the system. A user should not be able to access the system without adding a valid identity to the wallet.

Threat-6: Token acquisition used in the proposed incentive mechanism may be insufficient to motivate data providers. In order for the incentive mechanism to be established using tokens to motivate the roles in the best way, the amount of token earned and paid must be determined. In addition, determining the token value to represent a real-world asset will increase this effect.

7.4. Future Work

The infrastructure proposed in this thesis is specific to software project information, but it can be adapted for other kinds of information sharing and storage problems as well. As specific to software estimation, this work can be beneficial for companies that need to make estimations with software project data, and for organizations that want to make comparisons with software project information, like research centers, technology transfer offices etc.

In our upcoming work, by using the proposed model, a real system will be designed and implemented. Following that, an empirical study is planned with a research center in order to evaluate operational principles and validate the usability of the model in a broader context.

8. REFERENCES

[1] ISBSG, International Software Benchmarking standards Group, http://www.isbsg.org (Accessed: **16.01.2020**).

[2] PWC 15th Annual Global CEO Survey 2012, https://www.pwc.com.tr/en/publications/arastirmalar/pages/pwc-global-project-management-report-small.pdf (Accessed: **02.01.2020**).

[3] The Standish Group Chaos Report 2014, https://www.projectsmart.co.uk/white-papers/chaos-report.pdf (Accessed: **02.01.2020**).

[4] B. Kitchenham, E. Mendes, G.H. Travassos, Cross versus within-company cost estimation studies: A systematic review, In: Software Engineering, IEEE Transactions on, vol. 33, no. 5, pp. 316–329, **2007**.

[5] B. Turhan, E. Mendes, A Comparison of Cross-Versus Single-Company Effort Prediction Models for Web Projects. In: 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014. 285-292. 10.1109/SEAA.2014.41, **2014**.

[6] A. Idri, F. Amazal, A. Abran, Analogy-based software development effort estimation: A systematic mapping and review. In: Information and Software Technology. 58. 10.1016/j.infsof.2014.07.013, **2014**.

[7] J.M. Desharnais, Analyse statistique de la productivitie des projects informatique a partie de la technique des point des function, PhD thesis, Univ. of Montreal, **1989**.

[8] Desharnais dataset, http://promise.site.uottawa.ca/SERepository/datasets/desharnais.arff, (Accessed: **06.05.2019**).

[9] B.W. Boehm, Software Engineering Economics, Prentice Hall PTR, New Jersey, **1981**.

[10] COCOMO dataset, http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff, (Accessed: **06.05.2019**).

[11] QSM Software Project Database, https://www.qsm.com/resources/qsm-database, (Accessed: **23.06.2019**).

[12] QSM SLIM-Estimate, https://www.qsm.com/tools/slim-estimate, (Accessed: **23.06.2019**).

[13] QSM SLIM-Metrics, https://www.qsm.com/tools/slim-metrics, (Accessed: **23.06.2019**).

[14] B.W. Boehm, Software Cost Estimation Meets Software Diversity, In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, pp. 495-496, **2017**.

[15] T. Vera, S. Ochoa, D. Perovich, Survey of Software Development Effort Estimation Taxonomies, 10.13140/RG.2.2.14599.29601, **2018**.

[16] B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steece, Software Cost Estimation with COCOMOII, Prentice Hall, **2000**.

[17] J.W. Keung, B.A. Kitchenham, D.R. Jeffery, Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation, In: EEE Transactions on Software Engineering, vol. 34, no. 4, pp. 471-484, **2008**.

[18] M. Tsunoda, T. Kakimoto, A. Monden, K. Matsumoto, An Empirical Evaluation of Outlier Deletion Methods for Analogy-Based Cost Estimation, In: ACM International Conference Proceeding Series, **2011**.

[19] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf, (Accessed: **06.05.2019)**.

[20] Infographic: A Look at Blockchain Technology, https://burniegroup.com/infographic-a-look-at-blockchain-technology, (Accessed: **16.06.2019**)

[21] Ethereum, Blockchain App Platform, https://ethereum.org, (Accessed: **06.05.2019**).

[22] Hyperledger, Blockchain Technologies for business, https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf, (Accessed: **17.06.2019**).

[23] Hyperledger Fabric, A Blockchain Platform for Enterprise, https://hyperledger-fabric.readthedocs.io/en/latest/index.html, (Accessed: **16.12.2019**).

[24] Hyperledger Fabric Ledger, https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger/ledger.html, (Accessed: **02.01.2020**).

[25] Hyperledger Fabric Transaction Flow, https://hyperledger-fabric.readthedocs.io/en/release-1.4/arch-deep-dive.html, (Accessed: **02.01.2020**).

[26] Hyperledger Composer, https://hyperledger.github.io/composer/v0.19/introduction/introduction.html, (Accessed: **16.12.2019**).

[27] Hyperledger Composer Business Network Archive, https://hyperledger.github.io/composer/v0.19/introduction/introduction, (Accessed: **16.12.2019**).

[28] Hyperledger Composer Full-Stack Architecute, https://hyperledger.github.io/composer/v0.19/introduction/solution-architecture, (Accessed: **16.12.2019**).

[29] React, A JavaScript Library for Building User Interfaces, https://reactjs.org/, (Accessed: **16.12.2019**).

[30] jQWidgets, A JavaScript UI Framework, https://www.jqwidgets.com/, (Accessed: **16.12.2019**).

[31] A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, MedRec: Using Blockchain for Medical Data Access and Permission Management, In: 2nd International Conference on Open and Big Data (OBD) pp. 25-30, **2016**.

[32] J.P. Cruz, Y. Kaji, The Bitcoin Network as Platform for Trans-Organizational Attribute Authentication, In: The Third International Conference on Building and Exploring Web Based Environments (WEB), At Rome, Italy, **2015**.

[33] Q. Xia, E.B. Sifah, K.O. Asamoah, J. Gao, X. Du, M. Guizani, MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain, In: IEEE Access, vol. 5, pp. 14757-14767, **2017**.

[34] J.P. Cruz, Y. Kaji, N. Yanai, RBAC-SC: Role-Based Access Control Using Smart Contract, In: IEEE Access Volume 6, 12240-12251, **2018**.

[35] K. Liu, H. Desai, L. Kagal, M. Kantarcioglu, Enforceable Data Sharing Agreements Using Smart Contracts, https://arxiv.org/pdf/1804.10645.pdf, (Accessed: **06.05.2019**).

[36] K.R. Ozyilmaz, M. Doğan, A. Yurdakul, IDMoB: IoT Data Marketplace on Blockchain, In: Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, 2018, pp. 11-19, **2018**.

[37] V. Trón, A. Scher, D.A. Nagy, Z. Felföldi Z, N. Johnson, Swap, Swear and Swindle: Incentive system for Swarm. https://swarm-gateways.net/bzz:/theswarm.eth/ethersphere/orange-papers/1, (Accessed: **06.05.2019**).

[38] O.R. Kabi, V.N.L. Franqueira, Blockchain-Based Distributed Marketplace, In: Abramowicz W., Paschke A. Business Information Systems Workshops, Lecture Notes in Business Information Processing, vol 339, Springer, Cham, **2018**.

[39] N. Mittas, M. Athanasiades, L. Angelis, Improving analogy-based software cost estimation by a resampling method, Information & Software Technology, 50. 221-230. 10.1016/j.infsof.2007.04.001, **2008**.

[40] Y. Kamei,J. Keung, A. Monden,K. Matsumoto, An Over-sampling Method for Analogy-based Software Effort Estimation, 312-314. 10.1145/1414004.1414064, **2008**.

[41] M. Erhan, A. Tarhan, A. Ozsoy, A Conceptual Model for Blockchain-Based Software Project Information Sharing, In: IWSM-Mensura 2019, CEUR Proceedings, **2019**.

[42] C. Lokan, E. Mendes, Cross-company and single-company effort models using the ISBSG database: A further replicated study, 75-84. 10.1145/1159733.1159747, **2006**.

[43] Installing Hyperledger Composer, https://hyperledger.github.io/composer/latest/installing/installing-index, (Accessed: **16.12.2019**).

APPENDICES

Appendix 1 - Hyperledger Composer Installation

In this section Hyperledger Composer version 0.19 installation is explained. To install another version Hyperledger Composer Installation webpage [43] will help.

To run Hyperledger Composer on Ubuntu 16.0.4 the following are prerequisities for installing the development tools:

- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9 or higher (note version 9 is not supported)
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x

Hyperledger Composer provides a script file to install these prerequisites easly. It can be downloaded and run with following commands:

```
curl -O https://hyperledger.github.io/composer/v0.19/prereqs-ubuntu.sh

chmod u+x prereqs-ubuntu.sh

./prereqs-ubuntu.sh
```

After installing prerequisites, Hyperledger Composer can be installed with following commands:

1. To install essential CLI tools

```
npm install -g composer-cli@0.19
```

2. To install composer rest server

```
npm install -g composer-rest-server@0.19
```

3. To install utility for generating application assets.

```
npm install -g generator-hyperledger-composer@0.19
```

4. To install Yeoman tool.

```
npm install -g yo
```

5. To install Composer Playground.

```
npm install -g composer-playground@0.19
```

6. To install a local Hyperledger Fabric network.

```
mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers

curl -O https://raw.githubusercontent.com/hyperledger/composer-
tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz

tar -xvf fabric-dev-servers.tar.gz

cd ~/fabric-dev-servers

export FABRIC_VERSION=hlfv11

./downloadFabric.sh
```

Appendix 2 - Composer Rest Server Generation

Composer Rest Server is used to access the business network deployed to the Hyperledger Fabric network. The rest server is generated by the command provided by Hyperledger Composer. The steps required for this process are as follows:

- Admin card is imported to the business network.

```
composer card import --file networkadmin.card
```

- Demo participants are created using setupDemo transaction.

```
composer transaction submit -c admin@pis-network -d '{\"$class\":
\"org.pis.SetupDemo\"}'
```

- An identity card is issued for created participant.

```
composer identity issue -c admin@pis-network -f user1001 -u user1001 -a
org.pis.User#1001
```

- User identity card is imported to the business network.

```
composer card import --file user1001.card
```

- Rest server is started with user identity card.

```
composer-rest-server -c user1001@pis-network -p 3001
```

Appendix 3 - Business Network Archive Deployment

The commands provided by Hyperledger Composer are used to deploy Project Information Sharing Business Network. The steps to deploy implemented business network definition to Hyperledger Fabric network are given below. These steps assume that the development environment described in Appendix 1. is installed.

- Hyperledger Fabric is started.

```
export FABRIC_VERSION=hlfv11 && ~/fabric-dev-servers/startFabric.sh
```

- PeerAdmin is created.

```
export FABRIC_VERSION=hlfv11 && ~/fabric-dev-servers/createPeerAdminCard.sh
```

- Business network archive is created.

```
composer archive create --sourceType dir --sourceName . -a ./dist/pis-network.bna
```

- Business network archive is installed to the Hyperledger Fabric network.

```
composer network install --card PeerAdmin@hlfv1 --archiveFile dist/pis-network.bna
```

- Business network is started.

```
composer network start --networkName pis-network --networkVersion 0.0.19 --
networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1
--file networkadmin.card
```

Appendix 4 - Papers Derived From Thesis

M.Erhan, A.Tarhan, and A.Ozsoy, "A Conceptual Model for Blockchain-Based Software Project Information Sharing", in proceedings of The 29th International Workshop on Software Measurement (IWSM) and the 14th International Conference on Software Process and Product Measurement (MENSURA), IWSM-Mensura 2019, CEUR Proceedings.