**Requirements**

Workforce

Research

Guide

**Team: Group2**

Shin Yi Lin

Fenil Shah

Ramprasadh Srivathsa

Harshal Pawar

Mark Dwight

SE 6387.001

**ABSTRACT**

The elicitation and analysis of requirements is important and inevitable before starting to design. The client’s expectation on software functional and non-functional requirements may be different from developer’s thoughts. Also, different stakeholders may have different views and different expectations on the requirements. So, finding good ways to elicit and analyze requirements would facilitate the discussion between different stakeholders. Designers and developers won’t deviate from the goals.

**TABLE OF CONTENTS**

1. Introduction

1.1 Introduction to the document.....……………….……………………………………………………………5

1.2 Purpose and scope of the document..……………………………………………………………………..5

1.3 Description of the structure of the document ………………………………………………………...5

2. Use Case model for functional requirements

2.1 Use Case Model………………………………………………………………………………………………………..6

2.2 Use Case Description………………………………………………………………………………………………..7

3. Rationale for Use Case Model………………………………………………………………………………………………14

4. Non Functional Requirements.……………..…………..…………………………………………………………………15

5. Evidence that the requirements have been placed under configuration management….…….15

6. References……………………………………………………………………………………………………………………………15

**LIST OF FIGURES**

Figure 1: Use case diagram..……………….……………………………….……………….……………………………………6

**LIST OF TABLES**

Table 1: Add Fact/Belief use case description...……………….……………………………….………….……………7

Table 2: Batch adding of facts use case description.....…….……………………………….………….……………8

Table 3: Adding facts via folder scan use case description...............................................……………9

Table 4: Search Fact/Belief use case description..…..……….……………………………….………….………….10

Table 5: View Fact/Belief use case description..……..……….……………………………….………….………….10

Table 6: Delete Fact/Belief use case description.......……….……………………………….………….………….11

Table 7: Edit Fact/Belief use case description...……………….……………………………….………….………….11

Table 8: Login use case description................................................................................…………….12

Table 9: Add Fact/Belief meta-data item..................................................................…….……………13

Table 10: Delete Fact/Belief meta-data item................................................................................13

**1. INTRODUCTION**

**1.1 Introduction to the document**

This document describes the functional and nonfunctional requirements of the software Use case is the method to find the general views of functional requirements.  Performance, security, reliability, usability, and extensibility are the concerned nonfunctional requirements for the workforce research guide application.

**1.2 Purpose and scope of the document**

Requirement analysis helps clarify and discuss client’s needs, facilitates communication between different stakeholders, and guides them in right design direction. Also, it gives a blueprint for design in the next step. Without requirements, stakeholders may lose general goal and focus on detail part, which could make the whole application unreliable, unstable, etc.

**1.3 Description of the structure of the document**

The use cases are presented in the textual and graphical descriptions. In textual part, it includes use case names, IDs, participating actors, normal flow of events, exceptions, pre- and post- conditions, special requirements. In graphical parts, it includes the relationships between actors and use cases, and between different use cases.

There are 9 use cases: (1) user adds fact or belief, (2) user adds facts in batch, (3) user ads facts via folder scan, (4) user searches facts or beliefs, (5) user views facts or beliefs, (6) user updates facts or beliefs, (7) user deletes fact, (8) user adds dropdown list meta-data and (9) user deletes the dropdown list meta-data. A fact/belief sentence has attributes like country, metric, time-period , strength, etc., which helps in querying the data efficiently.

Based on the Workforce application’s prospected characteristics, performance, security, reliability, usability, and extensibility are important. User and administrator needs fast response in performance, account authentication and authority for data security, non-crushed system for reliability, easy-to-use app for usability, and easy-to-add functions and deployable in different platforms in future for extensibility.

**2. USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS**

**2.1 Use Case Model**

Following is the Use-case model of the Workforce Research Guide.

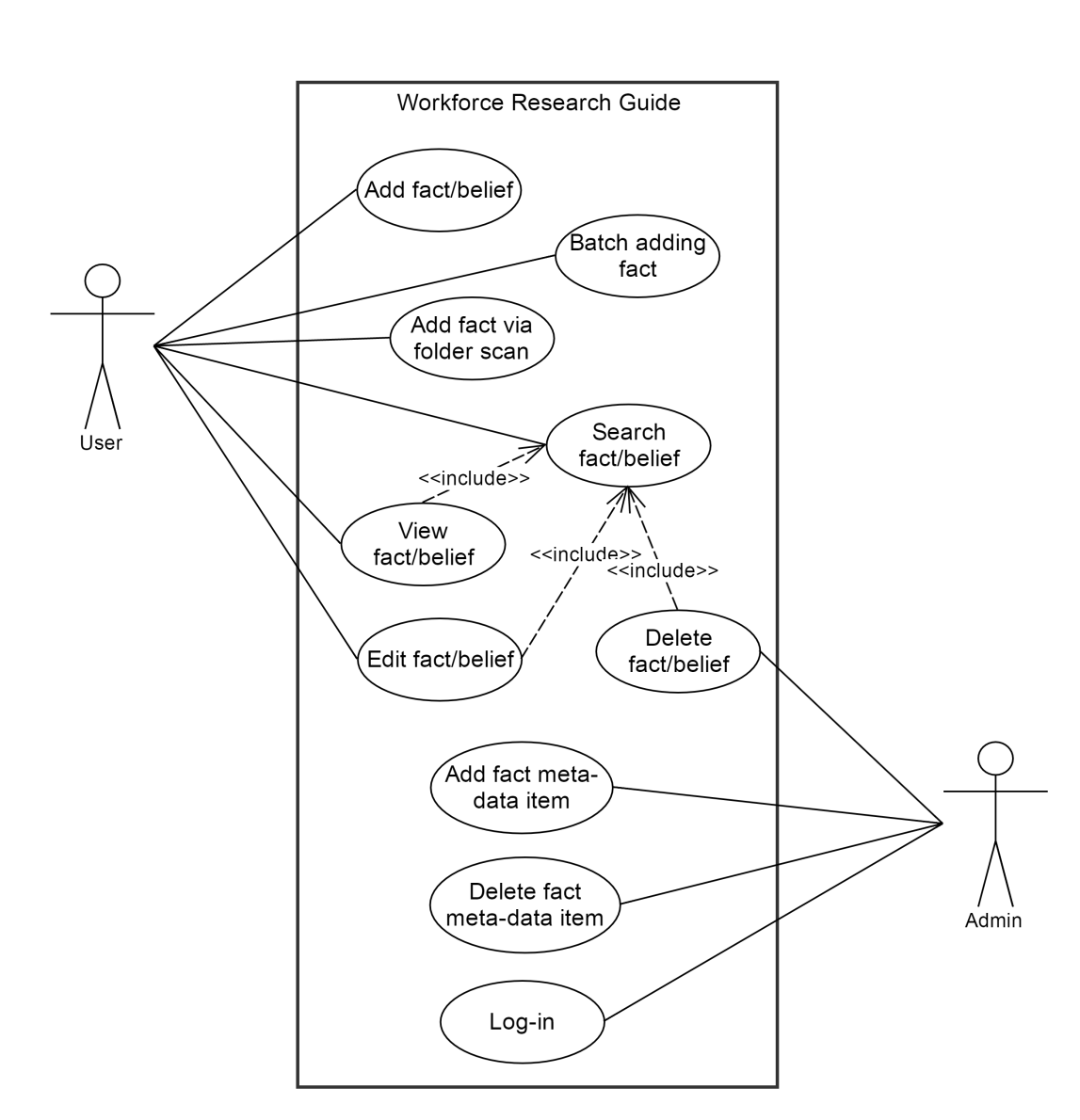


Figure 1 Use case diagram

**2.2 Use Case Description:**

|  |
| --- |
| **Use Case: Add Fact/Belief** |
| **ID**: 1 |
| **Brief Description**:  The user enters information about a fact or belief so that it can be stored by the software for future reference. |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:** |
| **Main Flow:**   1. The user selects the option to add a fact/belief. 2. The user enters information regarding that fact/belief (e.g. a name, description, country, time period, etc.) in all required fields 3. The user associates zero to many supporting documents for the fact/belief 4. The user submits the new fact/belief to the software |
| **Post-conditions:**  A new fact or belief is successfully added to the software’s collection of facts and beliefs |
| **Exceptions**:  The user selects to abort the addition before submitting. The software does not add a fact/belief and the use case ends. |

Table 1 Add Fact/Belief use case description

|  |
| --- |
| **Use Case: Batch Adding of Facts** |
| **ID**: 2 |
| **Brief Description**:  The user adds multiple facts at once by entering them via a batch file |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The batch file used is composed of facts only and is in .csv format |
| **Main Flow:**   1. The user selects the option to add a batch of facts 2. The user selects the batch file containing the facts to be added 3. The software adds all facts in the batch file to the software’s collection of facts and beliefs |
| **Post-conditions:**  All facts in the batch file have successfully been added to the software’s collection of facts and beliefs |
| **Exceptions**:  2.a. The user selects to abort the addition before submitting. The software does not add any facts and the use case ends.  2.b. The batch file is not formatted properly and is rejected by the software. The software does not add any facts and the use case ends. |

Table 2 Batch adding of facts use case description

|  |
| --- |
| **Use Case: Adding Facts via Folder Scan** |
| **ID**: 3 |
| **Brief Description**:  The user adds a fact and associates supporting documents with it all at once by selecting a folder |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The folder to be scanned exists on the user’s hard drive |
| **Main Flow:**   1. The user selects the option to add a fact via folder scan 2. The user selects the folder to be scanned 3. The software adds a fact with the same name as the scanned folder and associates all contained documents with that fact |
| **Post-conditions:**  A new fact is successfully added to the software’s collection of facts and beliefs. This new fact is associated with all documents contained in the user’s selected folder. |
| **Exceptions**:  2.a. The folder contains no documents and is rejected by the software. The user either selects another folder or exits the use case having added no new facts. |

Table 3 Adding facts via folder scan use case description

|  |
| --- |
| **Use Case**: **Search Fact/Belief** |
| **ID**: 4 |
| **Brief Description:**  Enables user to search for a fact or belief |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The software must be open |
| **Main Flow:**   1. User enters search items 2. User hits “Search” 3. Software looks for the items    1. If the search items are present, the software provides a list of possible facts or beliefs    2. If the search items are not present, the software informs the user that the items are not present |
| **Post-conditions:**  None. |

Table 4 Search Fact/Belief use case description

|  |
| --- |
| **Use Case: View Fact/Belief** |
| **ID:** 5 |
| **Brief Description:**  Enables user to view a fact or belief |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The software must be open |
| **Main Flow:**   1. Include (Search) 2. User selects the fact/belief to view. 3. User clicks on ‘View’. |
| **Post-conditions:**  The selected items are displayed to the user. |

Table 5 View Fact/Belief use case description

|  |
| --- |
| **Use Case: Delete Fact/Belief** |
| **ID:** 6 |
| **Brief Description:**  Enables user to delete a fact or belief |
| **Primary Actor:**  Admin |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The software must be open |
| **Main Flow:**   1. Include (Search) 2. Admin selects the fact/belief to delete 3. Admin clicks on ‘Delete’ |
| **Post-conditions:**  The selected item and its associated docs are deleted from the database/file. |

Table 6 Delete Fact/Belief use case description

|  |
| --- |
| **Use Case: Edit Fact/Belief** |
| **ID:** 7 |
| **Brief Description:**  Enables user to edit a fact or belief |
| **Primary Actor:**  User |
| **Secondary Actor:**  None |
| **Pre-conditions:**  The software must be open |
| **Main Flow:**   1. Include (Search) 2. User selects the fact/belief to edit. 3. User makes the changes and clicks on save |
| **Post-conditions:**  The changes made to the selected items are stored to the database/file. |

Table 7 Edit Fact/Belief use case description

|  |
| --- |
| **Use case**: Add fact/belief meta-data item |
| **ID**: 8 |
| **Brief Description**:  Admin adds new dropdown entry for fact/belief meta-data. |
| **Primary Actor**:  User |
| **Secondary Actor**:  None |
| **Pre-conditions**: |
| **Main Flow**:   1. User clicks on ‘Manage Dropdown data’ from the menu. 2. User clicks on ‘Add item’ on the screen. 3. User selects particular dropdown and enters entry into textbox. 4. User clicks on ‘Add entry’. |
| **Post-conditions**:  Newly added entry is available in that dropdown list. |

Table 8 Add Fact/Belief meta-data item use case description

|  |
| --- |
| **Use case**: Delete fact/belief meta-data item |
| **ID**: 9 |
| **Brief Description**:  Admin deletes existing dropdown entry for fact/belief meta-data. |
| **Primary Actor**:  User |
| **Secondary Actor**:  None |
| **Pre-conditions**:  Admin is logged in. |
| **Main Flow**:   1. User clicks on ‘Manage Dropdown data’ from the menu. 2. User clicks on ‘Delete item’ on the screen. 3. User selects particular dropdown. 4. The system will show all the entries for that dropdown. 5. User selects particular entry from the list and clicks on ‘Delete entry’. |
| **Post-conditions**:  Deleted entry is not available in that dropdown list. |

Table 9 Delete Fact/Belief meta-data use case description

**3. RATIONALE FOR YOUR USE CASE MODEL**

The greatest challenge a software developer faces is sharing the vision of the final product with the customer. All stakeholders in the project - developers, end users, software managers, customer managers - must achieve a common understanding of what the product will be and do, or someone will be surprised when it is delivered. Surprises in software are almost never good news. Therefore, we need ways to accurately capture, interpret, and represent the voice of the customer when specifying the requirements for a software product.

Often the customer will present as "needs" some combination of: the problems she has in her work that she expects the system to solve; the solutions she has in mind for an expressed or implied problem; the desired attributes of whatever solution ultimately is provided; and the true fundamental needs, that is, the functions the system must let her perform. The problem becomes more complex if the systems analyst is dealing with a surrogate customer, such as a marketing representative, who purports to speak for the actual end users of the application. The challenge to the analyst is to distinguish among these four types of input and identify the real functional requirements that will satisfy the real user's real business needs.

Many techniques are used for eliciting user requirements, all of which attempt to include the voice of the customer in the product design process. A typical project might employ a combination of meetings with user representatives and developers, facilitated workshops (for example, joint application design sessions) with analysts and users, individual customer interviews, and user surveys. The use case approach is an especially effective technique for deriving software requirements, analysis models, and test cases.

Each use case describes a scenario in which a user interacts with the system being defined to achieve a specific goal or accomplish a particular task. Use cases are described in terms of the user's work terminology, not computers. By focusing on essential use cases, stripped of implementation constraints or alternatives, the analyst can derive software requirements that will enable the user to achieve her objectives in each specific usage scenario.

For each use case, we stated the goal that the user needed to accomplish-one reason someone would use this application. Estimating the anticipated frequency of execution for each use case gave us a preliminary indication of concurrent usage loads, the importance of ease of learning versus ease of use, and capacities of data storage or transaction throughput. We could also identify the relative priority of implementing each use case at this stage. The sequence in which users identify candidate use cases suggests an approximate implementation priority.

By walking through individual use cases in the meetings, we drilled down to the fundamental customer needs the system had to satisfy. We also explored many "what if" scenarios to reveal exception and decision situations the system must handle.

**4. NON-FUNCTIONAL REQUIREMENTS**

**Performance**

The application shall provide high performance and we expect the query time to be around 5 seconds.

**Security and safety**

The application shall provide complete security for the user’s files and data. Also there would be a separate login for admin.

**Reliability**

The application shall be reliable in the terms that it will function as expected under circumstances like crashes, load balancing for a specified period of time.

**Usability**

The application shall provide usability by using simple, understandable graphical user interface. In this way, when a new user uses this app, s/he finds the application easy to use. Also, there will be separation of concerns between the user and the admin.

**Extensibility**

The application shall be implemented in such a way by modularizing the components so that it becomes easy for adding/removing functionalities.

**5. EVIDENCE THE REQUIREMENTS HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

<https://github.com/WorkforceResearchGuide/WorkforceResearchGuideApp.git>

**6. REFERENCES**

[1] http://www.agilemodeling.com/artifacts/useCaseDiagram.htm