Software Requirements Specification

for

Discussion Portal

Version 1.0 approved

Prepared by Ruchi Mehta

California Baptist University

September 3, 2018

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Project Scope 2

1.4 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. System Features 2

3.1 System Feature 1 2

3.2 System Feature 2 (and so on) 2

4. Data Requirements 2

4.1 Logical Data Model 2

4.2 Data Dictionary 2

4.3 Reports 2

4.4 Data Acquisition, Integrity, Retention, and Disposal 2

5. External Interface Requirements 2

5.1 User Interfaces 2

5.2 Software Interfaces 2

5.3 Hardware Interfaces 2

5.4 Communications Interfaces 2

6. Quality Attributes 2

6.1 Usability 2

6.2 Performance 2

6.3 Security 2

6.4 Safety 2

6.5 [Others as relevant] 2

7. Internationalization and Localization Requirements 2

8. Other Requirements 2

Appendix A: Glossary 2

Appendix B: Analysis Models 2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  | 9/9/18 |  | V0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to provide a detailed overview of software product, its parameters and goals. The document describes product user, target audience, user interface, hardware and software requirements. It will define how our client, team and audience see the product and its functionality.

## Document Conventions

## Project Scope

Cloud Based Discussion Portal allows students to share new ideas, allow peer to peer learning, where ideas can be implemented and lead to an invention. This Discussion portal will allow students to login and register for their account, place their questions, reviews question placed by other students and create a group for discussion. In cloud-based project the students will get real time update as any information is place by any other students- meaning users always have the most current information.

## References

# Overall Description

The overall objective of cloud-based discussion portal is to create connection between students of different field and share their ideas. The cloud-based discussion portal would be effective, efficient and user friendly and which:

* Increases the ability to create connection between student of different field.
* Allow user to bring different ideas together.
* Allow peer-to- peer learning.

## Product Perspective

The major reason in developing this system is to promote peer-to -peer learning

## User Classes and Characteristics

There are three types of user in this system that are students, faculty members and an administrator. Students and faculty member have same rules and regulations except one feature i.e only faculty members can create a group while students are not allowed to create one.

## Operating Environment

## Design and Implementation Constraints

The minimum requirement to access this system would be a user with PC that has internet connection and browser.

## Assumptions and Dependencies

It is assumed that students have a CBU email-id with which they would be authorized, they have browser in the system, internet connection and they are aware how does a website wok.

# System Features

## Login

### Description

System will allow different users to login into their account.

### Stimulus/Response Sequences

Each registered use can enter student email id and password to login to the system.

### Functional Requirements

Each user needs to enter valid student email id and password set for their account. On entering the correct user id and password the system will verify the data and user will be allowed to access the system. If the data entered is not correct an error message will be displayed stating to enter correct credentials.

## Register

### Description

New users can register themselves in the system and on authentication they can use the different feature of the system.

### Response Sequences

A registration form will be displayed on clicking the register now button.

### Functional Requirements

Registration form will have name, cbu id, degree type, department, cbu email id and set password are the mandatory fields that are required by the user to proceed further. ON entering the details, user will be allowed to login into the system using the user id and password provided in the form.

## Post on wall

### Description

After authentication the users can write post which will allow them to write any query or idea related to their upcoming projects and get help from different set of users.

### Response Sequences

The question or query will be posted on the wall.

### Functional Requirements

On clicking write a question tab user can write a question which will be posted on the wall.

## Set visibility

### Description

After posting question on the wall the user can set visibility for it.

### Response Sequences

On clicking set visibility a list of users will be displayed where user can search for specific set of users based on their major and username.

### Functional Requirements

The user needs to click on the set visibility button which will display a set of users based on their username and major.

## View post

### Description

Without posting any question user can view post made by them in past and view post of different users.

### Response Sequences

After logging into the system, the user can view the post on their wall.

### Functional Requirements

Once the user is logged in the system they just need to click on view post button to see post made by them in past and can see post of other users as well and sort them accordingly.

## Receive e-mail notification

### Description

System will send an email to the user who posted question if some response is made by some other user.

### Response Sequences

### User will be notified with an email which would say some user has given a response on the question posted by them.

### Functional Requirements

The user will automatically receiving an email there is not requirement for subscription.

## Delete user

### Description

Administrator can delete a user.

### Response Sequences

Based on passing year of students the administrator would delete the user only and not the post made by user.

### Functional Requirements

Administrator will have a list all students where passed out students would be deleted but the system will sill hold the data/post made by the user.

## Search user

### Description

User can search for different other users of the system.

### Response Sequences

A list of matching users will be listed based on the search.

### Functional Requirements

By writing username in the search tab user can find different students using the system.

## Set new password

### Description

User can set new password for their account.

### Response Sequences

The user will be asked to set new password for their account.

### Functional Requirements

User can click on forgot password button and new password can be set for their account.

## Create group

### Description

User/faculty member can create a group.

### Response Sequences

A user can join the group based on the invitation link received by them in the account.

### Functional Requirements

Faculty member can create a group based on their subject where students who are registered for the course would be sent an invitation to join the group and where question related to assignments can be posted by user and their response can be evaluated.

# Data Requirements

## Logical Data Model

<A data model is a visual representation of the data objects and collections the system will process and the relationships between them. Include a data model for the business operations being addressed by the system, or a logical representation for the data that the system itself will manipulate. Data models are most commonly created as an entity-relationship diagram.>

## Data Dictionary

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## Reports

<If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.>

## Data Acquisition, Integrity, Retention, and Disposal

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

# External Interface Requirements

<This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.>

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Software Interfaces

<Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.>

## Hardware Interfaces

<Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document.>

## Communications Interfaces

<State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.>

# Quality Attributes

## Usability

The software would be user-friendly, easy to use which would help students with their projects and with their assignments and homework. It will be peer-to peer learning process where students will help each other and help them to learn and grow together.

## Performance

The system needs to be very efficient as number of users are going to increase day by day so it will be such that user will not be kept waiting after performing some operation.

## Security

The software is only authorized to CBU students and faculty where their data would be safe any information related to project would not be seen by any unauthorized user. Students and faculty with CBU email id will only be allowed to use the system.

## Safety

The major goal of developing the system is to promote peer-to peer learning so the system is safe, secure and reliable to use.

# Internationalization and Localization Requirements

# Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models