Study Buddy

Software Requirements Specification

Version 3

5/7/2018

Software Engineers:

Andrew Medina

Jahaira Alaniz

Marco Olivares

Mauricio Azcona

Prepared for

Software Engineering

Instructor: MK Quweider, Ph.D.

Spring 2018

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 4/6/2018 | Creation/Original | All | First Publication |
| 4/20/2018 | Added Classes | All | Second Publication |
| 5/7/2018 | Final Edits | All | Final Publication |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  | Software Eng. |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**Revision History**](#_b3fuautni0t4) **2**

[**Document Approval**](#_q24847oy2o4y) **2**

[**1. Introduction**](#_88bbg46h6rxh) **5**

[**1.1 Purpose**](#_4at6dgr8qnzn) **5**

[**1.2 Scope**](#_6ymhrvv9vb7p) **5**

[**1.3 Definitions, Acronyms, and Abbreviations**](#_16tnok8i0esp) **5**

[**1.4 References**](#_fdzutktir7nf) **5**

[**1.5 Overview**](#_74h5km72p9pe) **6**

[**2. General Description**](#_wb7dr1jzqijo) **6**

[**2.1 Product Perspective**](#_e7bcbpsx3d1g) **6**

[**2.2 Product Functions**](#_viinfg1zs8n6) **6**

[**2.3 User Characteristics**](#_qaoqq79ihyzy) **7**

[**2.4 General Constraints**](#_kjgys8oho316) **7**

[**2.5 Assumptions and Dependencies**](#_20zyhaydbne1) **7**

[**3. Specific Requirements**](#_w0jzuam3p01m) **7**

[**3.1 External Interface Requirements**](#_5f2snpspmyba) **7**

[**3.1.1 User Interfaces**](#_xbgedj4azrmy) **8**

[**3.1.2 Hardware Interfaces**](#_al8rrtm0a730) **8**

[**3.1.3 Software Interfaces**](#_jbkbu89optcr) **8**

[**3.1.4 Communications Interfaces**](#_o02d5j9h0mqe) **8**

[**3.2 Functional Requirements**](#_1y810tw) **8**

[**3.2.1 Account Creation**](#_a7a4e5kixdpi) **8**

[**3.2.2 Account login**](#_oi89zycs75p8) **9**

[**3.3 Use Cases**](#_m3ghhoff3seu)

**3.3.1 Uuser Creates An Account**

**3.3.2 User Login**

**3.3.3 User Creates Group**

**3.3.4 User**

**12**

[**3.4 Classes / Objects**](#_7bh2v0z4wfc1) **12**

[**3.4.1**](#_3as4poj) **Database 12**

[**3.4.2**](#_jceq16kclbdp) **Student**

**3.4.3 StudentDataModel**

**3.4.4 GroupDataModel**

**3.4.5 Group**

**3.4.6 SBEmail**

**3.4.7 SBAddFriend**

**3.4.8. SBCreateGroupGUI**

**3.4.9 SBLoginGUI**

**3.4.10 SBMainGUI**

**3.4.11 SBNewEmailFormGUI**

**3.4.12SBNewAccountFormGUI**

**3.4.13 SBcontroller 13**

[**3.5 Non-Functional Requirements**](#_l1x5nedkcfez) **13**

[**3.5.1 Performance**](#_bxnunmg3az9j) **13**

[**3.5.2 Reliability**](#_7hesl7t7xl2f) **13**

[**3.5.3 Availability**](#_nwrkqfir62at) **13**

[**3.5.4 Security**](#_ezew9wccp8pb) **13**

[**3.5.5 Maintainability**](#_grww7gpe25ry) **13**

[**3.5.6 Portability**](#_mpyb3t4lmt11) **13**

[**3.6 Inverse Requirements**](#_1hmsyys) **13**

[**3.7 Design Constraints**](#_6sfkzpe6n0u2) **13**

[**3.8 Logical Database Requirements**](#_d4z5zpzgxf09) **14**

[**3.9 Other Requirements**](#_iveyhgkg09ge) **14**

[**4. Analysis Models**](#_lms7ybatmn42) **14**

[**4.1 Sequence Diagrams**](#_wtz2rtslj7k9) **14**

[**4.3 Data Flow Diagrams (DFD)**](#_4f1mdlm) **14**

[**4.2 State-Transition Diagrams (STD)**](#_nahyc9lvuwoz) **14**

[**5. Change Management Process**](#_19c6y18) **15**

# **1. Introduction**

*This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.*

## **1.1 Purpose**

*The purpose of this document is to give a detailed description of the requirements for the “Study Buddy” software. It will illustrate the purpose and complete declaration for the development of this software product. Furthermore, it will also explain in further detail the system constraints, interface and interactions with other external applications. This document is primarily intended to be viewed by the development team but will also be later viewed by the customer for approval purposes.*

## **1.2 Scope**

The “Study Buddy” app is a matching application that will match users (students) with one another. The students matched will have the same course and same instructor. The goal is to make it a mobile application. Although a valid internet connection is needed, the “Study Buddy” application will mainly require a database connection to look up other students that not only have the same course, but preferably, have the same instructor as well.

## **1.3 Definitions, Acronyms, and Abbreviations**

*USER - any person who is using the software*

*ADMIN -System administrator who is given specific permission for managing and controlling the system*

*DB - short for Database*

*STUD - (Abbreviation) student, which is the intended audience for this application*

*CRSE - (Abbreviation) course, which is another field that will be used to match students*

*PROF - (Abbreviation) professor, which is another field that will be used to match students*

*DOW- (Abbreviation) Days of Week , this will be both used to determine which days of week are the course, and what Days of the Week the student is free. This will be in the MTWRF format*

*CTIME -(Abbreviation) Course Time, which is the time of the Students course.*

## **1.4 References**

## **1.5 Overview**

*The remainder of this document includes three chapters and appendices. Section 2 provides an overview of the system functionality and system interaction with other systems. Section 2 also mentions the system constraints and assumptions about the product. Section 3 provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. Section 4 deals with the prioritization of the requirements. The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.*

# **2. General Description**

*In this Section of the SRS for “Study Buddy” it will give an overview of the whole system. The system will be explained in its context to show how the system interacts with the other external/software/hardware systems and introduce the basic functionality of it. It will also describe what type of users that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.*

## **2.1 Product Perspective**

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

*StudyBuddy will function as platform for students to find and join study groups. StudyBuddy will function and look similar to other social networking apps. Instead of feeds or threads, students will join a study group. Users in the same group will be referred to as study buddies.*

## **2.2 Product Functions**

*This subsection of the SRS should provide a summary of the functions that the software will perform.*

*StudyBuddy will let users:*

*(account management)*

1. *Create an account*
2. *Request a password change*
3. *Suspend their account*
4. *Delete their account*
5. *Login*
6. *Logout*

*(study group management)*

1. *Search for a study group*
2. *Start a new study group*
3. *Join a study group*
4. *Leave a study group*

*(communication)*

1. *Communicate with study buddies*
2. *Schedule study sessions via a calendar*
3. *upload study material*
4. *download study material*

## **2.3 User Characteristics**

*StudyBuddy’s users are college students. They are looking for other students to study with. The users should be familiar with other social networks/applications.*

## **2.4 General Constraints**

*Due to the fact that StudyBuddy fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function. The mobile application will be constrained by the capacity of the database. Since the database is shared between both applications, it may be forced to queue incoming requests and therefore may reduce the speed in general of the application .*

## **2.5 Assumptions and Dependencies**

*One assumptions about the Study Buddy application is that every user(student) that uses it will have in their possession a mobile phone that meets the minimum hardware requirements (to be determined later).*

# **3. Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

## **3.1 External Interface Requirements**

*This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.*

### **3.1.1 User Interfaces**

*The User Interface will start with a login screen asking for username and password.if the user is creating an account for the first time a “Create New Account” or similar option will be displayed. If the user hasn’t made an account yet, they will be required to do so, before then can access all of the features.*

### **3.1.2 Hardware Interfaces**

*Due to the fact that the mobile application does not have any designated hardware, it does not have any direct hardware interfaces at the moment.*

### **3.1.3 Software Interfaces**

*Since a web portal is required in our software, the Software Interfaces in this case would be between the web portal and the database.*

### **3.1.4 Communications Interfaces**

*Since the only communication with the application is with database and web portal, if a messaging/chat system is implemented then communication with mobile devices within the application would be required*

## **3.2 Functional Requirements**

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

### **3.2.1** Account Creation

3.2.1.1Introduction:

The account creation will allow the user to be able to create a new StudyBuddy account if they do not have one already so that can be able to use the applications main features.

3.2.1.2 Inputs:

The system will require three inputs from the user for account creation, which are:

username -must be between 2 to 25 characters, and not empty

password - must be between 2 to 25 characters, and not empty

email address - cannot be empty

3.2.1.3 Processing

The system will store the information to the database so that the software will remember the three input fields when the login is required again

3.2.1.4 Outputs

The system will print a message saying that the account was successfully created and that the user can return to the login screen with their new credentials

3.2.1.5 Error Handling

invalid/blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

invalid/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

### **3.2.2** Account login

3.2.2.1 Introduction:

The system will allow for the user the login using their credentials (username,password, email)

and will have an option to select a university so that they can search for sessions/buddies based on the school .

3.2.2.2 Inputs:

The system will require two inputs from the user for account creation, which are:

username -must be between 2 to 25 characters, and not empty

password - must be between 2 to 25 characters, and not empty

3.2.2.3 Processing

The system will search the database for existing username and passwords that match a student

3.2.2.4 Outputs

The system will print a message saying that the login was successful then will transfer the user to the main home application screen

3.2.2.5 Error Handling

blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

invalid username; if the password entered does not match an existing one in the database then the system will notify the user that the password entered does not matching any existing one

invalid password; if the password entered does not match an existing one in the database then the system will notify the user that the password entered does not matching any existing one

**3.2.4 Account logout**

3.2.4.1Introduction

The system will allow for the user to logout which will take them takc to the main login menu

3.2.4.2 Inputs:

The system will have an option/button that will allow for the user to logout and go back to the login screen

3.2.4.3 Processing

The system will go back to the login screen once the button has been pressed

3.2.4.4 Outputs

The system will print a message saying that the user has logged out

3.2.4.5 Error Handling

N/A - the system should not allow for errors when logging out

**3.2.5 Account Deletion**

3.2.5.1 Introduction:

The system will allow for the user to delete their studybuddy account.

3.2.5.2 Inputs:

The system will require the user to enter their username and password to verify that it is indeed the right user that wishes to delete their account

3.2.5.3 Processing

once the correct username and password is given the system will go ahead and erase the username and password from the database

3.2.5.4 Outputs

The system will print a message saying that the account was successfully deleted

3.2.5.5 Error Handling

invalid/blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

invalid/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

**3.2.6 Password reset**

3.2.6.1 Introduction:

The system will allow for the user to reset their password if they forgot the original will they created or wish to create a new one

3.2.6.2 Inputs:

The system will require the username and the email of the account that they wish to create the new password for

3.2.6.3 Processing

Once the correct username and email is entered, the system will allow for the user to create the new password that they wish to do

3.2.6.4 Outputs

The system will print a message saying that the new password was successfully created

3.2.6.5 Error Handling

invalid/blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

invalid/blank emeil: if the email that was entered was blank or isn't the correct one belonging to the user then a message will display saying invalid email.

invalid/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

**3.2.7 Study group search**

3.2.7.1 Introduction:

The system will allow for the user to search for a study group session that has alrady been created by another user within the same university

3.2.7.2 Inputs:

The system will require the user to enter the course id and it will bring up all sessions that have that course number as well as their respective instructors

3.2.7.3 Processing

The system will allow for the option for the user to be able to join the session

3.2.7.4 Outputs

The system will print a message saying that user has successfully join the session

3.2.7.5 Error Handling

invalid/blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

invalid/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

**3.2.8 Study group creation**

3.2.8.1 Introduction

The system will allow for the user to create a study group session

3.2.8.2 Inputs:

The system will require the user to enter the name of the session that they wish to crate, and the course as well as the instructor for the session

3.2.8.3 Processing

The system will store the information to the database and will show up for users who search for either the course or the instructor

3.2.8.4 Outputs

The system will print a message saying that study group was successfully created

3.2.8.5 Error Handling

Not much error handling will occur unless the user creates a session that has a ame that already exists

invalid/blank username: if the the username field is left blank then the system will print an error message saying that there is an empty field

invalid/blank password: if the password field is left blank then the system will print an error message saying that there is an empty field

**3.2.9 File upload/download**

3.2.9.1 Introduction

The system will allow for the user upload files which will help the study session in one way or another

3.2.9.2 Inputs:

the input will be the File Chooser button which will prompt the user for the file to select

The system will require the user to enter the name of the session, the

3.2.9.3 Processing

once the file is selected, it will be shared with the other users in the session

3.2.9.4 Outputs

The system will state the the file was successfully uploaded

3.2.9.5 Error Handling

**3.2.10 View Members of Study Group**

3.2.9.1 Introduction

The system will allow for the user to view the members of the studygroup one they have seleted the specific group

3.2.9.2 Inputs:

the input will be the selection of the study group itself

3.2.9.3 Processing

one the group is selected, the system will pull from the database all of the users that are connected to that group besides the current user

3.2.9.4 Outputs

The system will show all of the members of that group

3.2.9.5 Error Handling

## **3.3 Use Cases**

### 

## **3.4 Classes / Objects**

3.4.1 Database

|  |  |
| --- | --- |
| **Class Class Name:** Database  **Brief Description:** default class that initializes database connection, statement, and the result set from the JDBC  **Attributes (fields)**  **Attribute Description**  Connection SBConn  Name of the JDBC Connection for the database that is going to be used for StudyBuddy which cannot be overridden as the connections/ports are static  Statement SBStmt  Default JDBC Statement that can be overridden in calls from the application to generate an sql statement for the database  ResultSet SBRs  The Default JDBC result set that that can be overridden in calls from the application to generate an sql statement for the database  **Methods (operations)**  **Method Description**  <methdbConnect()  Initializes the attributes and attempts the connection to the specified database and schema  Database()  Private Constructor method for Database class, calls dbConnect() to initialize the connection  getSingletonofDatabase  Public method that accesses the private database constructor | |
|
|
|
|
|
|
|
|
|

3.4.2: **Student.java**

|  |  |
| --- | --- |
| **Class Name:** Student  **Brief Description:** the class that contains the student object, which is one of the main objects that StudyBuddy uses throughout the application  **Attributes (fields)**  **Attribute Description**  Integer sid  The unique student id that identifies the student  String firstname  The first name for the student  String lastname  The last name for the student    String email  The email for the student  String major  The student’s current major for the school they are studying at  **Methods (operations)**  **Method Description**  Student(String, String, String, String String ,int)  constructor for the student class | |
|
|
|
|
|
|
|
|
|

3.4.4 StudentDataModel.java

|  |  |
| --- | --- |
| **Class Name:** StudentDataModel  **Brief Description:** Holds the datamodel for the student  **Attributes (fields)**  **Attribute Description**  Database mydb  Database for storing users information  ArrayList <Student> stulist  ArrayList for storing a list of students  Student student  Student object for user information organization  Database mydatabase  Database for functionality testing | |
|
|
|
|
|
|

3..4.5 GroupDataModel.java

|  |  |
| --- | --- |
| **Class Name:** GroupDataModel | |
| **Brief Description:** Model class that stores group information | |
| **Attributes (fields)** | **Attribute Description** |
| ArrayList<Group> grouplist | ArrayList for storing list of groups |
| Group group | Group object for organizing group information |

3.4.6 Group.java

|  |  |
| --- | --- |
| **Class Name:** Group  **Brief Description:** Class that stores the group object  **Attributes (fields)**  **Attribute Description**  String gname  private string for the group name  int gid  private int for the group id  String gsubject  private string for the group subject  String daysofweek  private string for the days of the week  int courseid  private int for the course id  String currentuser  private string for the current user    **Methods (operations)**  **Method Description**  Group(String gname, int gid, String gsubject, String ginstructor, int courseid)  public constructor for the group class | |
|
|
|
|
|
|
|
|
|
|
|

3.4.7 SBEmail.java

|  |  |
| --- | --- |
| **Class Name:** SBEmail  **Brief Description:** email class that contains the email function  **Attributes (fields)**  **Attribute Description**  String host  private string name for the email provider host.  String port  private string name for the email provider port.  String mailFrom  private string name for the email address from the sender.  String password  private string name for the email password from the sender.  String mailTo  private string for the email address from the receiver.  String subject  private string for the email subject  String message  private string name for the email message.  String attachFiles  private string name for the email attachment string.  String ccEmails  private string name for multiple email receivers  singletonOfDatabase  private static for the Singleton    **Methods (operations)**  **Method Description**  setSBEmail(String host, String port, String mailfrom, String password, String mailTo, String subject,  String message ,String[] attachFiles  constructor for the email to be send to user. | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.4.8 SBAddFriend.java

|  |  |
| --- | --- |
| **Class Name:** SBAddFriend  **Brief Description:** GUI class that has the form to add friends  **Attributes (fields)**  **Attribute Description**  Label friendlabel  private label for the username to be entered.  TextField friendTF  private textfield for the user to enter the friends he/she wants to add  Button addfriendBTN  private button for the user to add a friend  Label statuslabel  private label for the status    **Methods (operations)**  **Method Description**  SBAddFriend()  Initializes the attribute to add a friend | |
|
|
|
|
|
|
|
|
|

3.4.9 SBCreateGroupGUI.java

|  |  |
| --- | --- |
| **Class Name:** SBCreateGroupGUI  **Brief Description:** <description>  **Attributes (fields)**  **Attribute Description**  Label gidLBL  private label for the user to enter the group ID.  Label gnameLBL  private label for the user the group name  Label cidLBL  private label for the user to enter the course ID number.  Label subLBL  private label for the user the subject.  Label dowLBL  private label for the user to enter the days of the week  TextField gidTF  private textfield where the user enters the group ID  TextField gnameTF  private textfield where the user enters the group name  TextField cidTF  private textfield where the user enters the course ID number  TextField subTF  private textfield where the user enters the subject  TextField ginstructorTF  private textfield where the user enters the days of the week    **Methods (operations)**  **Method Description**  void SBCreateGroupGUI()  public void for the group when be created.  void clearfieldgroup()  public void for the group field to be cleared. | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.4.10 SBLoginGUI.java

|  |  |
| --- | --- |
| **Class Name:** SBLoginGUI | |
| **Brief Description:** The login GUI form | |
| **Attributes (fields)** | **Attribute Description** |
| Button CreateAccountBTN | private button for the user to create account |
| Button LoginBTN | private button for the user to login |
| Button ForgotBTN | private button for the user to have access to the link when forgetting the password. |
| Label UserNameLBL | private label for the username |
| Label PasswordLBL | private label for the password |
| Label SchoolLBL | private label for the school |
| TextField UserNameTF | private textfield for the username |
| PasswordField PasswordTF | private textfield for the password |
| VBox LoginVB | private vbox for the login option. |
| HBox ButtonHB | private hbox button for Hbox. |
| Label Title | private label for the GUI title |
| Label CopyWrite | private label for out team logo |
| VBox titlebox | private vbox for the title |
| String currentuser | private string for the current user. |
| ImageView iv | image container for displaying an image |
| Image image | university image |
|  | |
| **Methods (operations)** | **Method Description** |
| clearfieldslogin(TextField tf1, Password tf2) | clears username and password fields |
| SBLoginGUI() | Initializes attributes |

3.4.11 SBMainGUI.java

|  |  |
| --- | --- |
| **Class Name:** SBMainGUI  **Brief Description:** holds the main home screen view for the user  **Attributes (fields)**  **Attribute Description**  Button FriendSearchBTN  private button for the user to search a friend  Button AddFriendBTN  private button for user to add a friend.  Button HideFriendBTN  private button for user to hide a friend  Button CreateAppBTN  private button for the user to create a new appointment  Button LogoutBTN  private button for the user to logout from the app.  Button UploadFileBTN  private button for user to upload a file  Button DoneBTN  private button for the user to press finish  Button addGroupBTN  private button for the user to add a group.  Button createGroupBTN  private button for user to create a new group  Button refreshGroupBTN  private button for the user to refresh their group.  Label NewFriendLBL  private label to display “New friend”  TextField NewFriendTF  private textfield for the new friend label  Label WelcomeLBL  private label to display “Welcome”  Label Friends  private label to display “Friends”  Label Session  private label to display “Session”  Label OnlineSes  private label to display “Online Session”  Label AddFriendSLabel  private label to display add a friend  Label AnnouncementsLabel  private label to display “Announcements”  TextArea AnnouncementTA  private text area for the announcements field  TextArea DisplayFriendTA  private text area for the display friend field  TextArea DisplaySessionTA  private text area for the display session field  ImageView iv  private Image view that will be display in the GUI  Image image  This will display the image attached  HBox Apphbox  HBox for the display of the group table and the create group button  VBox Appvbox  VBox for the display of CreateApp button, upload file button, logout button, friends, slist, friend search button  HBox Welcomevbox  the vbox that has the title and school image  ListView<String> slist  listview that stores the list of students for the friends list  ListView<String> glist  listview that stores the list of groups from the groups list  ListView<String> mylist  a new instantiation of listview  TableView grouptable  the table that displays the groups  String fvalue  dynamic string that stores the selected string when it is selected from the friend slist  TableColumn groupname  table column for group table that holds the name of the group  TableColumn groupid  table column for group table that holds the group id of the group  TableColumn groupcid  table column for group table that holds the course id of the group  TableColumn groupsub  table column for group table that holds the name of the group  TableColumn groupinst  table column for group table that holds the instructor for the group  final Label tablelbl  the label for the table itself    **Methods (operations)**  **Method Description**  SBMainGUI()  Initializes attributes | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.4.12 SBNewEmailFormGUI.java

|  |  |
| --- | --- |
| **Class Name:** SBNewEmailFormGUI  **Brief Description:** Email GUI that shows when the user is creating a new email  **Attributes (fields)**  **Attribute Description**  private Label etoLBL  private label to display “Email To:”  private Label esub  private label to display “Email Subject”  private Label emessage  private label to display “Email Message:”  private Label fileLBL  private label to display “Please enter filename you wish to send:”  private TextField etoTF  private textfield for the email to label  private TextField esubTF  private textfield for the email subject label  private TextArea emessageTF  private textarea for the email message label  private TextField fileTF  private textfield for the file being send label.  private Button eSubmitBTN  private button for the submusion button    **Methods (operations)**  **Method Description**  SBNewEmailForm()  Initializes attributes | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.4.13 SBNewAccountFormGUI.java

|  |  |
| --- | --- |
| **Class Name:** SBNewAccountFormGUI  **Brief Description:** GUI that holds the view when the user creates a new account  **Attributes (fields)**  **Attribute Description**  Button CreateAccBTN  private button to create new account  Button exitBTN  private button to exit current site.  Label fname  private label for users first name  Label lname  private label for the users last time  Label uname  private label for the users username  Label pass  private label for the users password.  Label email  private label for the users email.  Label slabel  private label for the user to enter their campus from the options provided.  Label majorlabel  private label for the users to enter their degree major.  RadioButton MButton  private button when the users selection of gender male  RadioButton FButton  private button for the user to their selection as gender female  Label NATitle  private label that displays the title of the form.  Label SBid  private label that displays “School ID”  TextField SBidTF  private textfield for the school id label  TextField fnameTF  private textfield for the first name label.  TextField lnameTF  private textfield for the last name label  TextField unameTF  private textfield for the username label  TextField passTF  private textfield for the password label  TextField emailTF  private textfield for the email label  TextField majorlabelTF  private textfield for the degree major label  HBox RButtonhbox  private HBox is the cotainer    **Methods (operations)**  **Method Description**  SBNewAccountFormGUI()  private constructor method for the new account form class | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.4.14 SBController.java

|  |  |
| --- | --- |
| **Class Name:** SBController  **Brief Description:** main controller that controls all the elements of the application  **Attributes (fields)**  **Attribute Description**  Stage primary stage;  public stage for the class primary stage.  SBMainGUI mGUI  This allows the SBMainGUI to display  SBCreateGroupGUI sbcreategroup  displays the SBCreateGroupGUI class  SBNewEmailForm sbcreateemail  displays the SBNewEmailForm class  StudentDataModel sdm;  displays the StudentDataModel class  SBLoginGUI lGUI ;  displays the SBLoginGUI class  SBNewAccountFormGUI snafGUI;  displays the SBNewAccountFormGUI class  SBMain main;  displays the SBMain class  SBAddFriend sbaf;  displays the SBAddFriend class  String currentuser ;  private string for the current user  GroupDataModel gdm1  displays theGroupDataModel class  GroupController gcontroller;  displays the GroupController class  ObservableList olist  displays the ObservableList class  Image utrgvimage  displays the image attached for the class that will display when the user enters utrgv as their campus option.  ImageView utrgviv  displays the image attached for the class that will display when the user enters utrgv as their campus option  Image tscimage  displays the image attached for the class that will display when the user enters tsc as their campus option  ImageView tsciv  displays the image attached for the class that will display when the user enters tsc as their campus option  Image stcimage  displays the image attached for the class that will display when the user enters stc as their campus option  ImageView stc  displays the image attached for the class that will display when the user enters stc as their campus option  Alert errorAlert  displays an alert message when user selects or enters an option not taken by the system.  **Methods (operations)**  **Method Description**  SBController(SBMainGUI maingui, StudentDataModel studentdm, SBLoginGUI sblgui, SBNewAccountFormGUI sbnagui, SBAddFriend sbaf)  MVC Controller for connecting GUI elements with model  void attachHandlers()  Relays user interactions from GUI to model | |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

3.1.15 EmailAttachmentSender.java

|  |  |
| --- | --- |
| **Class Name:** EmailAttachmentSender  **Brief Description:** email class that contains the methods for sending the emails  **Methods (operations)**  **Method Description**  void sendEmailWithAttachments(String host, String port, final String userName, final String password, String toAddress, String subject, String message, String[] attachFiles)  sends an email with an attachment  void sendEmailWithAttachmentsCC(String host, String port, final String userName, final String password, String toAddress, String subject, String message, String[] attachFiles,String[] ccEmails)  sends a mass email with an attachment to users | |
|
|
|
|

…

## **3.5 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

### **3.5.1 Performance**

*The Application should be fast enough as long as the resources it requires are not allocated to different mobile applications in the background to provide the best experience.*

### **3.5.2 Reliability**

*The Application should be able to give the correct information when looking up courses, students, so that it would provide false data.*

### **3.5.3 Availability**

*Although it is impossible for the software to be up and running 100% of the time, the goal is to be available more than at least 98% of the time*

### **3.5.4 Security**

*The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.*

### **3.5.5 Maintainability**

*The Application should be written/coded in a way that will allow for being able to easily implement new functions/features when necessary.*

### **3.5.6 Portability**

*The Application will be portable in the sense that both Android and iOS users will be able to use the software as intended.*

## **3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

## **3.7 Design Constraints**

*Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

## **3.8 Logical Database Requirements**

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

## **3.9 Other Requirements**

*Catchall section for any additional requirements.*

# **4. Analysis Models**

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

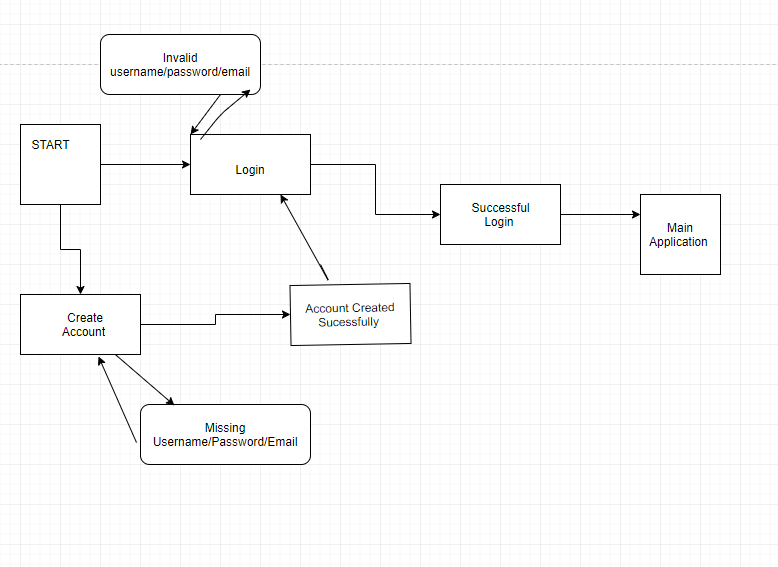
## **4.1 Sequence Diagrams**

Since the application is still in the planning phase, we have do not have enough information that will allow us to create diagrams. The next revision of this SRS will have them surely.

## **4.3 Data Flow Diagrams (DFD)**

## **4.2 State-Transition Diagrams (STD)**

4.2.1. Main STD



# **5. Change Management Process**

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

## 