Software Design Documentation

StudyBuddy

Study Group Platform

Version 3.0

May 7, 2018

Lead Software Engineers:

Jahaira Alaniz

Andrew Medina

Project Team:

Marco Olivares

Mauricio Azcona

Prepared for

Software Engineering

University of Texas Rio Grande Valley

Instructor: MK Quweider, Ph. D.

Spring 2018

Table of Contents

1. Introduction…………………………………………………………………………………29
   1. Purpose…………………………………………………………………………………
   2. Scope…………………………………………………………………………………
   3. Definitions, Acronyms, and Abbreviations……………………………………………………………………………
2. References…………………………………………………………………………………
3. Detailed Design…………………………………………………………………………………
4. User Interfaces by Use Case…………………………………………………………………………………
5. Appendices…………………………………………………………………………………

A.1 Appendix 1…………………………………………………………………………………

A.2 Appendix 2…………………………………………………………………………………

**1. Introduction**

**1.1 Purpose**

This Software Design Document is made with the purpose of explicitly outlining the software architecture and high level design of the StudyBuddy application. Through the use of various architectural decompositions, the various modules, data structures, databases and interfaces will be depicted. The intention of this document is to provide developers an insight into meeting our Client’s needs efficiently and effectively, corresponding with the requirements set forth in the SRS. Therefore, this document is mainly intended for the developers, present company as well as future organizations.

**1.2 Scope**

The software application described throughout this SDD is the StudyBuddy. This SDD is intended for a base level system in order to provide a proof of concept for the use of building an evolutionary prototype that demonstrates the functionality specified by the corresponding SRS. This will be achieved through the use of use-case models, state models, class models and data flow models that will serve to clarify the development teams thought process during implementation.

**1.3 Definitions, Acronyms, and Abbreviations**

The following terms, acronyms, and abbreviations are used throughout this document and are presented in the table below by order of appearance.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SDD | Software Design Description; A written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. |
| SRS | Software Requirement Specification; A comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform |
| StudyBuddy |  |
| GUI | Graphical User Interface; provides a visual, interactive means for a software user to manipulate the controls, commands, or features of that software. |
| Wizard | A sequential set of prompts for input, assisting in data collection and organized such that its implementation increases ease of use. |
| Database | A structured collection of data that can be efficiently and conveniently accessed |
| JDO | Java Data Object; A standard way to access persistent data in databases, using plain old Java objects (POJO) to represent persistent data. |
| POJO | Plain Old Java Object; An ordinary Java object, not bound by any special restriction |
| IDE | Integrated Development Environment; software that provides tools for the development and organization of programming code. |
| JVM | Java Virtual Machine; Provides the necessary links allowing a java program to run on a machine using a particular operating system. |
| JRE | Java Runtime Environment; Including the Java Virtual Machine, all necessary components for a system to establish the environment in which Java programs will run. |
| DBA | Directs or performs all activities related to maintaining a successful database environment |
| SQL | Structured Query Language; the standard relational database query language. |
| JDBC | Java Database Connectivity; a Java API developed by Oracle Corporation which provides methods for querying and updated a database. |

**2. References**

Git - <https://git-scm.com/>

GitHub - <https://github.com/>

HTML - <http://www.w3schools.com/html/>

Java Virtual Machine - <https://java.com/en/download/>

Java Runtime Environment - <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

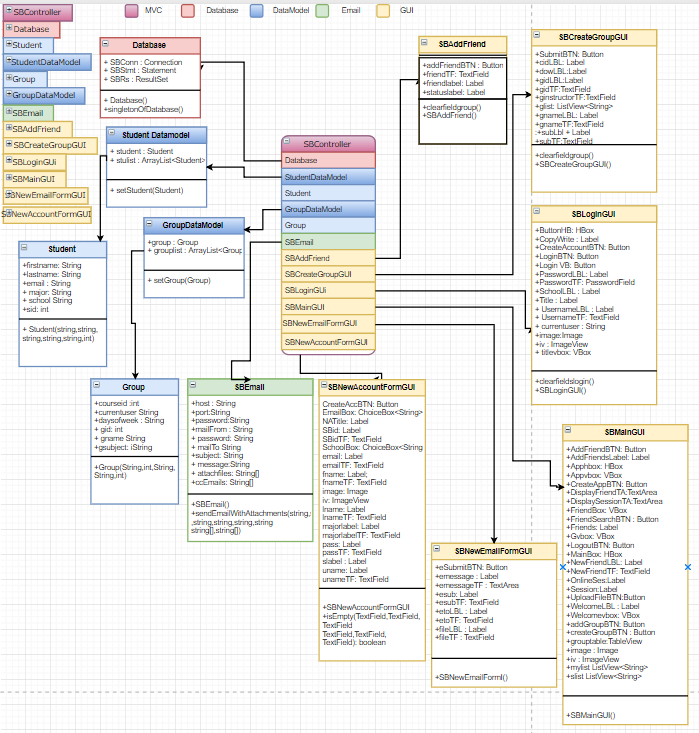
JavaFX - <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

MySQL - <http://dev.mysql.com/downloads/mysql/>

NetBeans - <https://netbeans.org/>

3. Detailed Design

3.1 Module Detailed Design



3.1.01: **Database. java**

|  |  |
| --- | --- |
| **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 |

3.1.02: **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 |

3.1.03 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 |

3.1.04 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 |

3.1.05 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 |

3.1.06 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. |

3.1.07 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 |

3.1.08 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. |

3.1.09 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 |

3.1.10 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 |

3.1.11 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 |

3.1.12 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 |

3.1.13 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 |

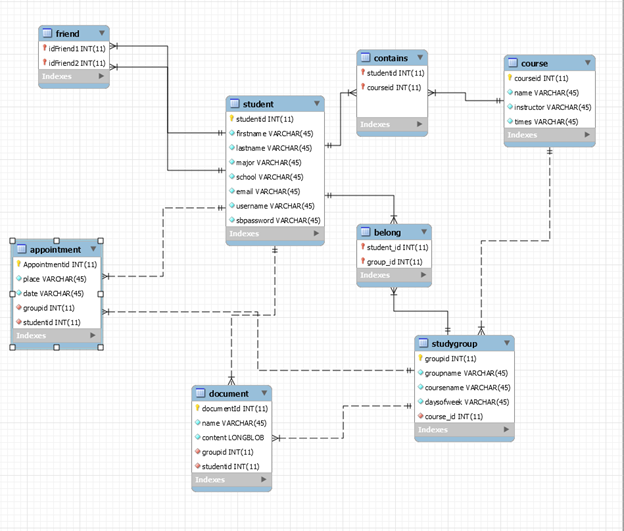
3.1.14 GroupController.java

|  |  |
| --- | --- |
| **Class Name:** GroupController | |
| **Brief Description:** controller that only focuses on group functions and datamodels | |
| **Attributes (fields)** | **Attribute Description** |
| Group group | Group object for organizing group information |
| ArrayList<Group> grouplist | ArrayList of groups |
| SBCreateGroupGUI creatgroupGUI | SBCreateGroupGUI object for controller |
| SBMainGUI sbmain | SBMainGUI object for controller |
| SBLoginGUI sblogin | SBLoginGUI object for controller |
| GroupDataModel gdm | GroupDataModel object for controller |
| ObservableList<Group> oGroup | ObservableList to display groups |
|  | |
| **Methods (operations)** | **Method Description** |
| GroupController( GroupDataModel gdmm, SBCreateGroupGUI cg, SBMainGUI smain) | MVC controller for connecting GUI elements with the 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 |

**3.2 Data Detailed Design**



3.2.01 Student

|  |  |
| --- | --- |
| **Table: Student** | |
| **Description** | It contains all the information of the users. |
| **Attributes** | **Description** |
| studentId | Student Id, int, primary key, not null. AutoIncrements |
| firstname | First name, varchar(45), not null |
| lastname | Last name, varchar(45), not null |
| major | Major, varchar(45), not null |
| school | School, varchar(45), not null |
| email | Email address, varchar(45), not null |
| username | Username for the application, varchar(45), not null |
| sbpassword | Password of the account, varchar(45), not null |

3.2.02 Document

|  |  |
| --- | --- |
| **Table: document** | |
| **Description** | It contains files uploaded by the users. |
| **Attributes** | **Description** |
| groupid | Id for the study group, primary key, int, not null AutoIncrements |
| groupname | name of the group, varchar(45), not null |
| coursename | name of the course, varchar(45), not null |
| daysofweek | Appointment days, varchar(45), not null |
| courseid | Id of the course, int, foreign key. |

3.2.03 Studygroup

|  |  |
| --- | --- |
| **Table: studygroup** | |
| **Description** | All the information of the study groups |
| **Attributes** | **Description** |
| documentid | Id for the document, primary key, int, not null |
| name | name of the document, varchar(45), not null |
| content | Content of the file, large BLOB, not null |
| groupid | Id of the group, int, foreign key |
| studentid | Id of the course, int, foreign key. |

3.2.04 belong

|  |  |
| --- | --- |
| **Table: belong** | |
| **Description** | shows the relationship of users and groups they belong to |
| **Attributes** | **Description** |
| student\_id | Id for the user, int ,foriegn key for student.studentid |
| group\_id | Id for the group, int, foreign key for studygroup.groupid |

3.2.05 Friend

|  |  |
| --- | --- |
| **Table: friend** | |
| **Description** | shows the relationship between 1 user and another |
| **Attributes** | **Description** |
| idFriend1 | id for the first user friend relationship , int, foreign key for student.studentid |
| idFriend2 | d for the second user relationship , int, foreign key for student.studentid |

3.2.06 Contains

|  |  |
| --- | --- |
| **Table: contains** | |
| **Description** | Shows what students belong to which course |
| **Attributes** | **Description** |
| studentid | Id for the user, int ,foreign key for student.studentid |
| courseid | Id of the course, int, foreign key for course.courseid |

3.2.07 Course

|  |  |
| --- | --- |
| **Table: course** | |
| **Description** | Stores all the info about a course |
| **Attributes** | **Description** |
| courseid | Id for the course,primary key, int, not null |
| name | name of the course varchar(45), not null |
| instructor | name of the instructor for the course, varchar(45), not null |
| times | Days of the week (MW) or (Tth) of the course, varchar(45) not null |

3.2.08 Appointment

|  |  |
| --- | --- |
| **Table: Appointment** | |
| **Description** | All the information for the appointments |
| **Attributes** | **Description** |
| Appointmentid | Id for the appointment,primary key, int, not null |
| place | location of appointment, varchar(45), not not null |
| date | The date of the appointment, varchar(45), not null |
| groupid | Id of the group, int, foreign key, not null |
| studentid | Id of the course, int, foreign key., not null |

4. User Interfaces by use case scenarios

4.1

**Use Case 01: User creates a new StudyBuddy account.**

StudyBuddy will require that users create a new account if they do not have one already.

Valid Scenario

1. User clicks on create account button.

2. User successfully inputs a valid username and studentid.

3. User clicks on create account button.

4. Alert message prompts the user saying that their account was created successfully.

4. User clicks on accept button.

Invalid Scenario 1

1. User clicks on create account button.

2. User inputs a username that already exists.

3. Alert prompts the user that the username is already taken.

4. User remains at screen until issue is fixed.

Invalid Scenario 2

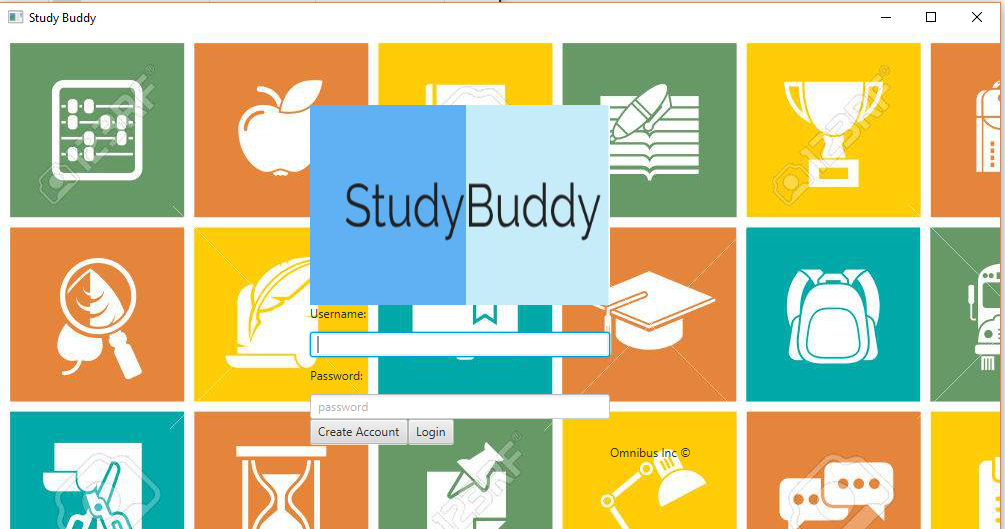
1. User clicks on create account button.

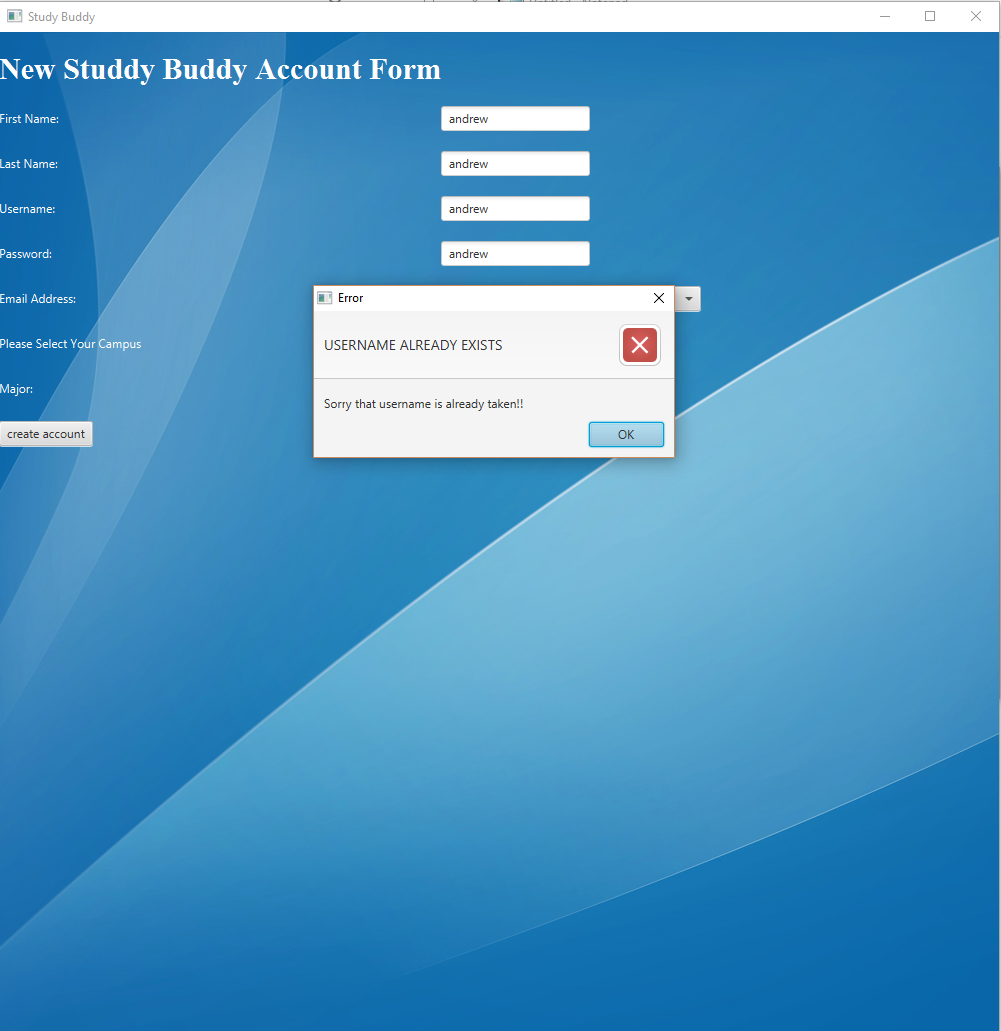
2. User leaves a field blank.

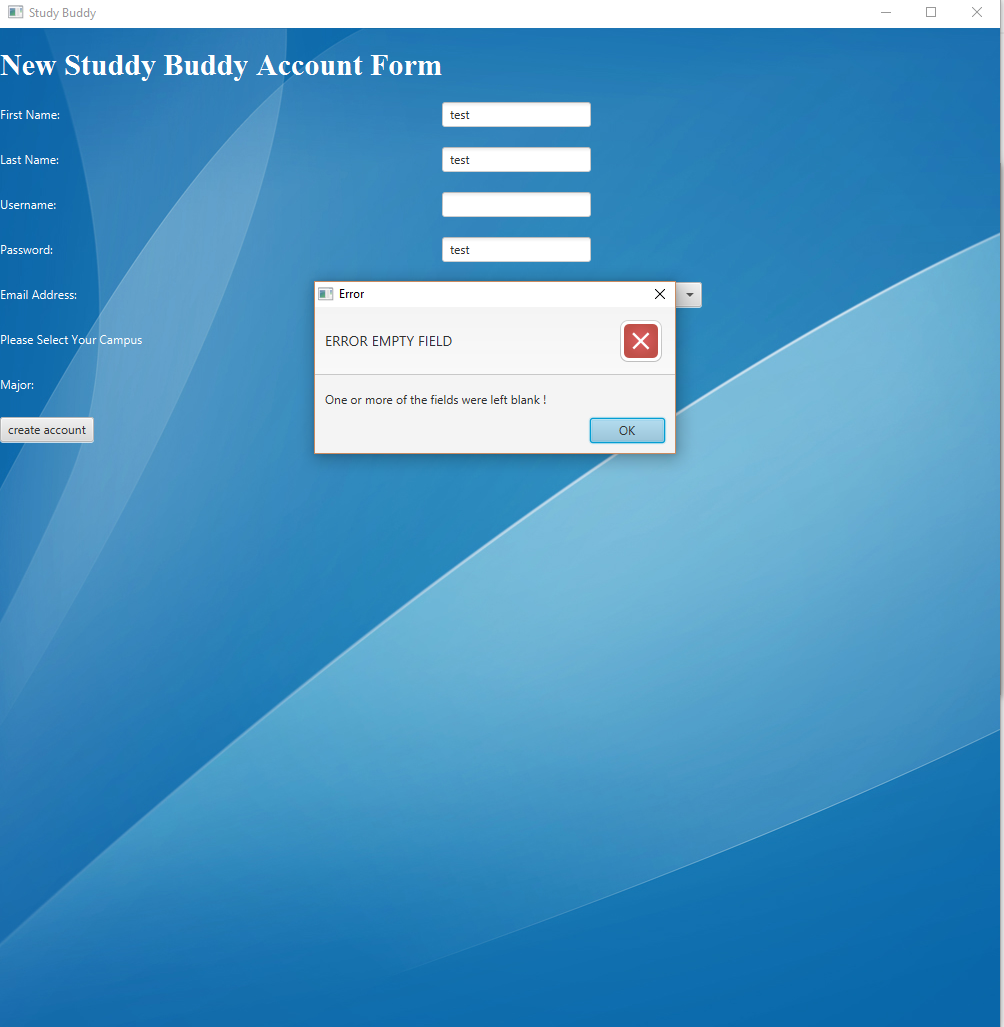
3. Alert prompts the user that one of the fields was blank

4. User remains at screen until issue is fixed.

**Case 01.**

****

****

****

4.2

**Use Case 02: User Login**

StudyBuddy will allow the user to login once they have an existing account

A Valid Scenario

1. The user enters the correct username and password

2. The user clicks on login button.

2. The user is now at the StudyBuddy main screen.

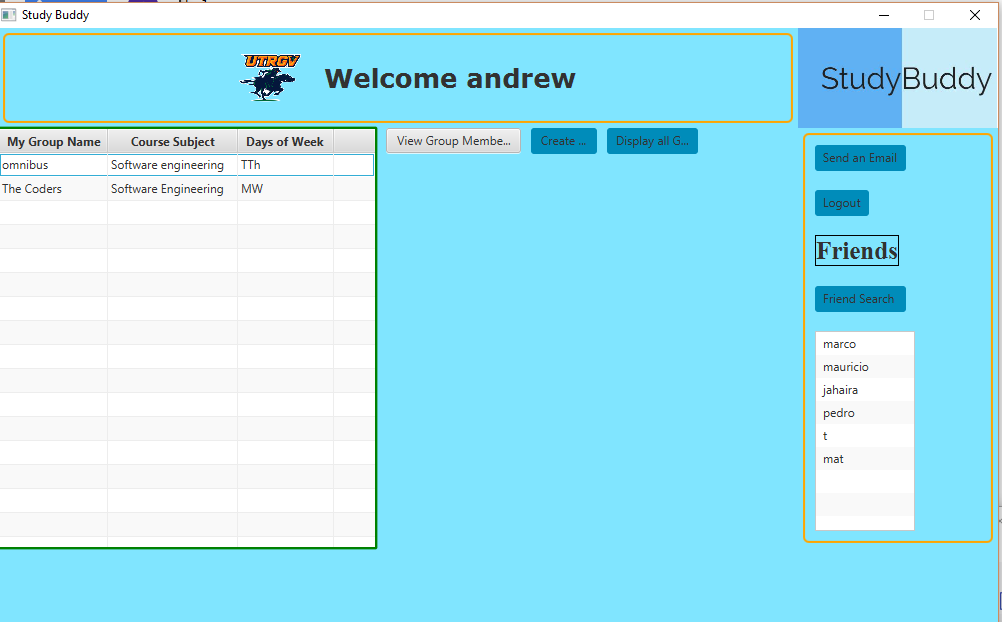
Invalid Scenario 1

1. The user enters and invalid username or password.
2. Alert message will prompt the user saying that an invalid username/password has been entered.

Invalid Scenario 2

1. The user leaves either username/password field blank or both fields blank
2. An alert message prompts the user saying that one of the fields was left blank

**Case 02.**

****

****

****

4.3

**Use Case 03: User creates a new group.**

StudyBuddy will allow the user to create a new study group.

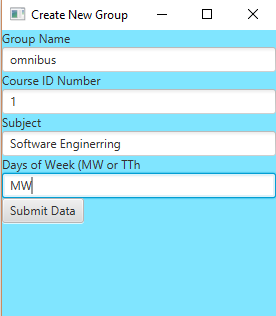
Valid Scenario

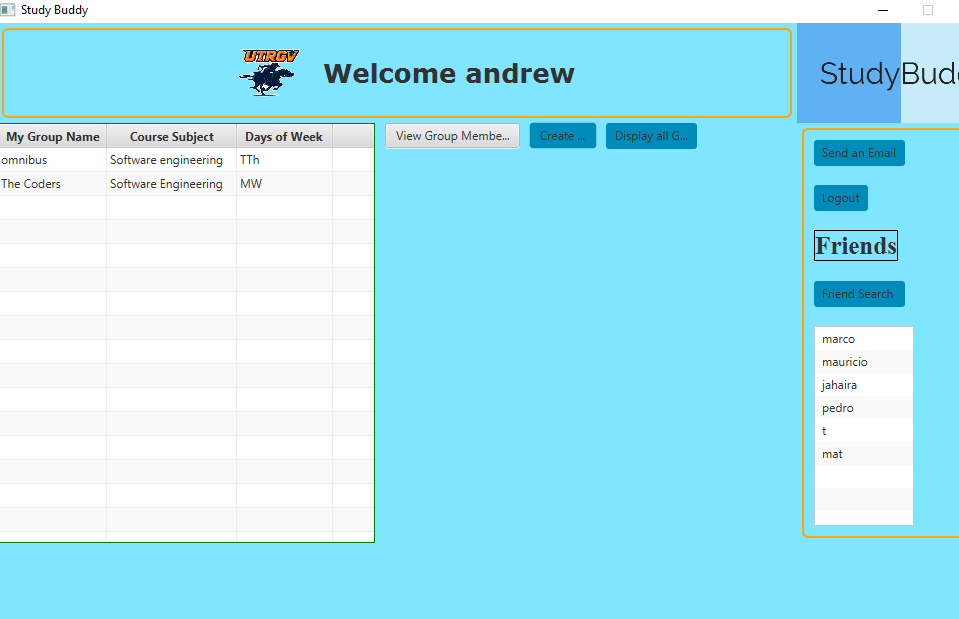
1. User clicks on create group button in StudyBuddy main screen.
2. User provides the required information.
3. User clicks on submit data button.
4. User is able to see the new group in StudyBuddy main screen.

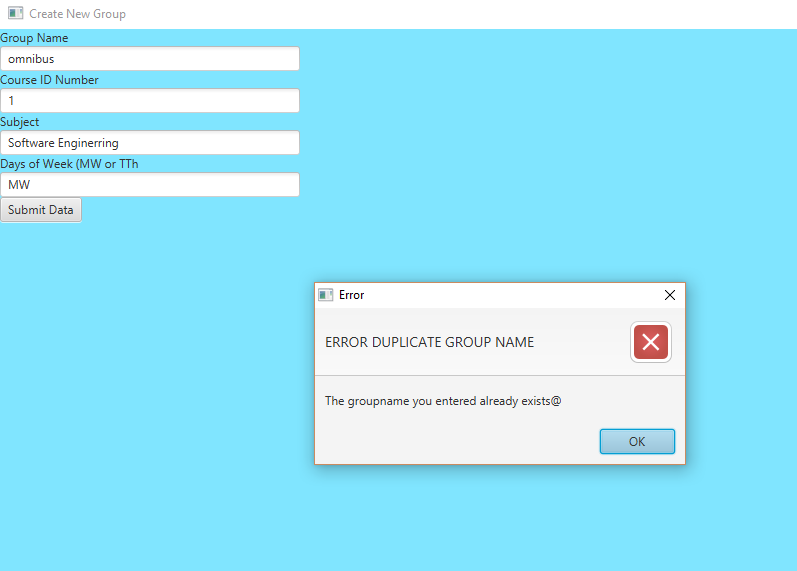
Invalid Scenario

1. User clicks on create group button
2. User enteres an existing group name
3. Alert message prompts user saying that the groupname already exists
4. User remains in menu till issue is fixed

**Case 03.**

****

****

****

4.4

**Use Case 04: User adds another user as friend.**

StudyBuddy will allow the user to ask another user to be friends.

Valid Scenario

1 User clicks on friend search button in StudyBuddy main screen.

2 User provides the username of the student.

3 User clicks on add friend button.

4 User can see the new group in StudyBuddy main screen.

Invalid scenario 1

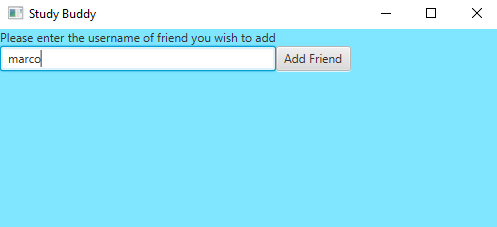
1 User clicks on friend search button in StudyBuddy main screen.

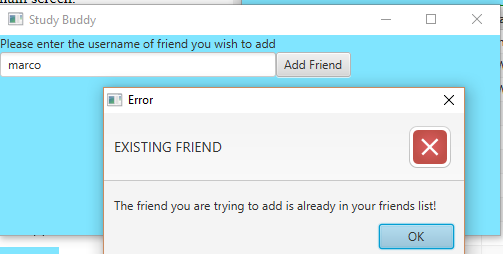
2 User provides a name that is already in friends list

3 User clicks on add friend button.

4 An alert message prompts the user saying that that person is already a friend

**Case 04.**

****

****

4.5

**Use Case 05: User sends an email.**

StudyBuddy will allow the user to send emails with attached files.

Valid Scenario

1 User clicks on the friend they wish to send the email

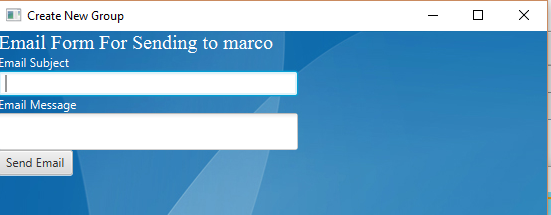
2 User clicks on send an email

3 User inputs subject and message

4 Email is sent.

.

**Case 05.**



4.6

**Use Case 06: User finishes his session in StudyBuddy.**

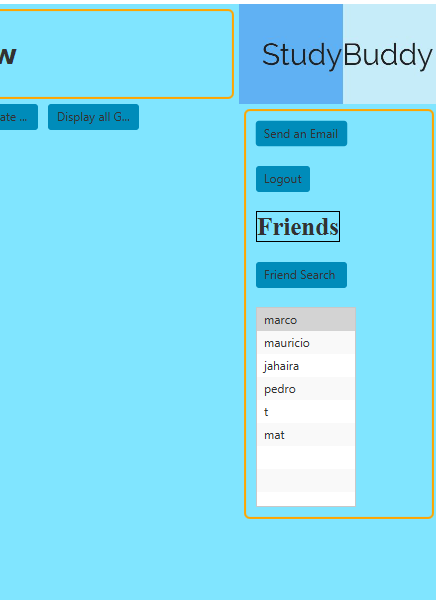
StudyBuddy will allow the user to finish the current session whenever he wants.

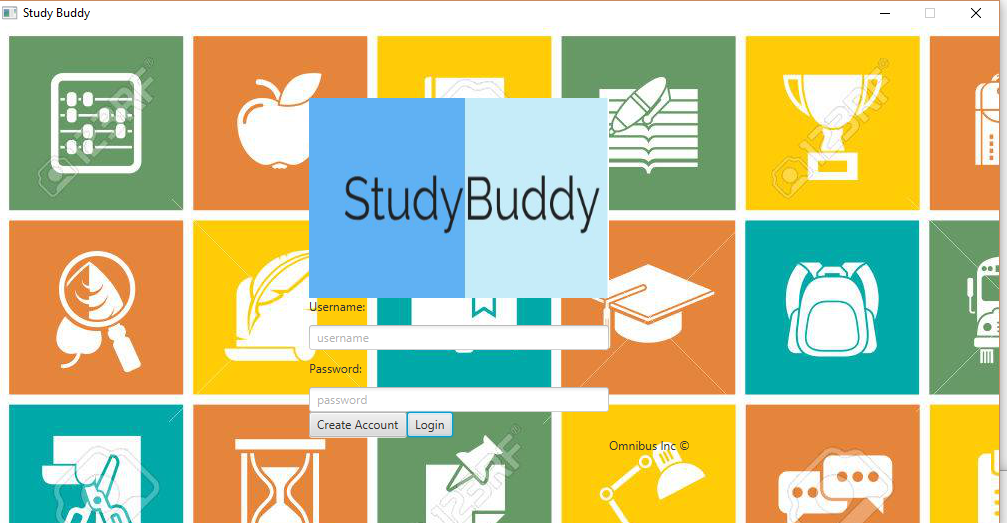
Valid Scenario

1 User clicks on logout button in StudyBuddy main screen.

2 User is sent to login screen.

**Case 06.**

****

****

A. Appendices

A.1 Appendix 1

Required form for the approval of changes to this SDD document:

Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Jahaira Alaniz | Lead Software Engineer |  |
|  | Andrew Medina | Lead Software Engineer |  |
|  | Dr. M.K. Quweider | CSCI 3340 Instructor |  |

A.2 Appendix 2

Required form must be attached to the end of this document if there are any changes after its

initial completion:

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Comments |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |