CS 411W Lab 2, Version 2
Product Specification
Thy Barsell, Gold Team
1 April, 2019
1 Introduction

Section 1 will cover the purpose; scope; definitions, acronyms, and abbreviations; and references of Study Buddy.

1.1 Purpose

Study Buddy will be a web application that will act as a platform that helps students and people pursuing new knowledge find the perfect study groups to achieve academic success. As an application, Study Buddy’s core feature is to match people with others based on set preferences and filters. Individual availability, study subject, learning styles, and group size, among other attributes, will be used to find potential study buddies for someone. The application will have integrated tools for groups to use for online collaboration.

1.2 Scope

Study Buddy will serve as a medium for people to match with others who are like-minded in learning styles and study preferences to study for specific topics or subjects together. This will effectively connect people together, especially students who typically have trouble finding classmates to study with. The intended goal is to create a platform for people to quickly find ideal study partners to work together with, boosting socialization and productivity.

As a prototype application, Study Buddy’s core feature is still to match people with others based on set preferences and filters. The prototype will also feature limited integrated API applications, such as Google Hangouts.
1.3 Definitions, Acronyms, and Abbreviations

**Auditory Learner** - Someone who best comprehends information by listening to information rather than reading it or seeing it visually.

**Business Logic** - The programming that manages communication between an end user interface and a database.

**CRUD** - Stands for Create, Read, Update, and Delete. Basic database/application operations.

**Entity Class** - A simple Java Class with member variables and getter and setter methods defined.

**JPA** - Java Persistence Application programming interface is an API for handling all database operations such as storing or retrieve entities from the database.

**JSF** - Java Server Faces is a Java framework that couples the view and servlet into one managed component.

**Kinesthetic Learner** - Someone who best comprehends information by participating in activities or solving problems in a hands-on manner.

**ORM** - Object-relational mapping. Technique for persisting objects into a database table. Tables are modeled after Entity classes.

**Procrastination** - Delaying or postponing a task, which needs to be completed, often to the detriment of the procrastinator.

**Prototype** - A reduced scale version of the final product that will demonstrate the functionality of the completed product in a simulated environment.

**Reading/Writing Learner** - Someone who best comprehends information by reading texts to further absorb information by condensing and rephrasing it in traditional lecture and note-taking environments.

**Study Group** - A small group of students with similar goals who meet regularly to review course material and prepare for exams.

**Visual Learner** - Someone who best comprehends information by visualizing relationships and ideas through maps, charts, diagrams and even essays.
1.4 References


Codeshare information, Retrieved September 12, 2018, from: https://codeshare.io/


Forgetting All Your Coursework, Retrieved September 16, 2018, from: https://www.blackboard.odu.edu/webapps/discussionboard/do/message?action=list_messages&course_id=_323209_1&nav=discussion_board_entry&conf_id=_351171_1&forum_id=_309142_1&message_id=_7348213_1


Slack Information, Retrieved September 12, 2018, from: https://slack.com/

Google Meet information, Retrieved September 12, 2018, from: https://support.google.com/a/answer/7303775?hl=en


2 General Description
Section 2 will cover the prototype architecture description, prototype functional description, and external interfaces of Study Buddy.

2.1 Prototype Architecture Description

The prototype product will be a web application accessible through web browser. Users will be able to create an account, set up a profile, and fill out a survey to determine their study preferences at registration. Users can then search for a buddy by entering their study topic, and an intelligent buddy matching algorithm derived from the user’s study preferences will populate a list of potential matches. From there, the matched users may communicate on the website via a messaging system, or move to using Google Hangouts to schedule video chat meetings.

Study Buddy (prototype) is comprised of the following major components:

- Login: allows user entry of authentication credentials
- New user: allows a user to create an account
- Study preference setting: allows users to set study preferences for intelligent buddy matching algorithm
- Search for buddies: allows user to search for study buddies
- Create a study group: allows user to create a study group
- Web application: the way in which the user will interact with the Study Buddy application, using a web browser
- Partner match by subject of interest: matching study buddies by their own subject interest
- Intelligent buddy matching: matching study buddies with the proprietary algorithm
- Google Hangouts integration: allows users to integrate their Google Hangouts accounts for setting meeting times and web conferencing

2.2 Prototype Functional Description

The major functional components of the Study Buddy prototype include the following:

- User account creation: An account system will allow for users to register and log into the application. Each account will have a public profile that users may choose to toggle privacy settings for when screening potential study partners. Users with registered accounts will also be required to complete a survey regarding their study preferences before being able to find a match.
- Intelligent buddy matching system: This matching system will use similarity scores to determine how well suited two individuals are for each other based on their study preferences. The matching system is the core feature of the application.
- Google Hangouts integration: External integration of a Google Hangouts API will allow users to collaborate in an environment that they may be more familiar with. Hangouts
will empower matched users to take the next step in communication through video chat meetings.

![Proposed Process Flow Diagram](image)

**Figure 1. Proposed Process Flow**

### 2.3 External Interfaces

#### 2.3.1 Hardware Interfaces

Study Buddy will not require the use of any hardware interfaces.

#### 2.3.2 Software Interfaces

Java Server Faces (JSF) will act as the foundation for the Study Buddy web application. As a framework, JSF allows for robust and modern user interface design, and also supports expression language to allow for easy data flow between the application and the server.

Java Persistence API (JPA) will solve any data persistence issues. JPA will supply the mechanisms required for SQL query construction by using an object-relational mapping technique to achieve object persistence.

#### 2.3.3 User Interfaces

Users will interact with the prototype product using any device that can access the Internet and has an up-to-date web browser. This includes personal computers, laptops, and mobile devices. Study Buddy will be mobile responsive.

### 3. Specific Requirements
This portion describes the specific requirements for the prototype version of Study Buddy which will be deployed as a web application. Each requirement is followed by a line in parenthesis that indicates who originally wrote the requirement and who edited the requirement. The format resembles the following:

(O: Last Name, M1: Last Name, M2: Last Name, … , MN: Last Name)
The O indicates the author who originally drafted the requirement and Mi indicates the ith author who modified the requirement.

3.1 Functional Requirements

3.1.1 Account Creation

3.1.1.1 The application shall allow users to create an account from the user interface.
(O: Carlson, M1: Best)

3.1.1.2 Once the application creates an account, the corresponding user shall receive an email confirmation.
(O: Carlson, M1: Myers)

3.1.1.3 The application shall notify users with an error message when invalid user credentials are submitted.
(O: Best)

3.1.1.4 The application shall allow two-factor authentication.
(O: Williams, M1: Hylton)

3.1.1.5 The application shall ask the user to complete a survey to determine their study preferences.
(O: Barsell)

3.1.1.6 The application shall allow the user to choose to not complete the study preferences survey at account creation, but at a separate time as well.
(O: Barsell)

3.1.1.7 The application shall allow the existing user to login from the login page and home page.
(O: Carlson, M1: Sharkey, M2: Barsell)

3.1.2 Account Management

3.1.2.1 The application shall allow logged in users to edit account information.
(O: Carlson)

3.1.2.2 The application shall allow logged in users to change their username.
3.1.2.3 The application shall allow logged in users to change their email address.

3.1.2.4 The application shall allow users to request a reset for their forgotten passwords.

3.1.2.5 When a password reset is requested, the user shall receive an email with instructions on how to securely reset their password.

3.1.2.6 The application shall allow users to be automatically logged in for up to 2 weeks before requiring manual login.

3.1.2.7 The application shall allow users to add buddies to the blocked buddies list.

3.1.2.8 The application shall allow users to remove buddies from the blocked buddies list.

3.1.2.9 The application shall allow users to edit their profile study preferences after completing the initial study preferences survey.

A sample list of study preferences shall include:

- **Learning Styles:**
  - Auditory - preferring to hear concepts to grasp information
  - Visual - preferring to see images, read text books to grasp information.
  - Kinesthetic - preferring to learn by doing to grasp information.

- **Communication Styles:**
  - Passive: don't often express opinions and communicate indifferently.
  - Aggressive: expresses opinions in a demanding manner, without regard for the other person.
  - Passive-Aggressive: passive externally with aggressive internal feelings during communication.
  - Assertive: communicates with purpose but without the demanding and overbearing attitude present in Aggressive communicators.

- **Study Styles:**
  - Social (interpersonal) - preferring to study/learn in groups or with another person.
  - Solitary (intrapersonal) - preferring to work alone or using self study.
Motivation Styles

- Intrinsic: includes things like: interest in the subject, curiosity, wanting to master the subject, self-motivation, enjoying learning.
- Extrinsic: includes things like: fear of punishment, monetary rewards, fear of failure, competition from others.

(O: Hylton, M1: Best, M2: Barsell)

3.1.2.10 The application shall provide users with the option to display or hide their personal information:

- Real name
- Profile picture
- Phone number
- Email address

(O: Sharkey, M1: Best, M2: Barsell)

3.1.3 Searching for buddies

3.1.3.1 The application shall allow a user searching for a buddy to choose a class/subject/topic at the time the search process is initialized.
(O: Barsell, M1: Best)

3.1.3.2 The application shall create an initial list of buddies who search for the same class/subject/topic.
(O: Sharkey)

3.1.3.3 The application shall remove buddies with different availability from the potential matched list.
(O: Sharkey)

3.1.3.4 The application shall remark blocked buddies whose names appear on the potential matched list.
(O: Sharkey)

3.1.3.5 The application shall calculate the similarity score between the searcher and a buddy using a cosine of an angle between 2 vectors where the vectors contain the users’ study preferences.
(O: Sharkey, M1: Myers, M2: Sharkey)
3.1.3.6 The application shall sort the list of potential matched buddies in descending order based on the similarity scores between the searcher and each buddy on the list.
(O: Sharkey, M1: Myers)

3.1.3.7 The application shall allow users to set a wait time limit for their desired end date of the search if:

- No matches are found
- The searcher does not decide to join existing groups
- The searcher only matches with buddies on the blocked list
(O: Sharkey, M1: Best, M2: Sharkey)

3.1.3.8 The application shall allow users to cancel the search if:

- No matches are found
- The searcher does not decide to join existing groups
- The searcher only matches with buddies on the blocked list
(O: Sharkey)

3.1.4 Forming study groups

3.1.4.1 The application shall allow the users to create a new study group by sending invitations to buddies on the matched list.
(O: Sharkey)

3.1.4.2 The application shall allow users to view contact information for matched buddies.
(O: Williams, M1: Sharkey, M2: Myers)

3.1.4.3 The application shall allow users to rate their previous study groups on a scale of 1-5.
(O: Fields, M1: Best, M2: Fields)

3.1.4.4 The application shall allow users to join existing study groups.
(O: Sharkey)

3.1.5 External APIs integration
3.1.5.1 The application shall utilize the third party API, namely Google Hangouts, for communication between users.
(O: Crotzer)

3.1.5.2 The application shall utilize the third party API, namely Slack, for communication between users.
(O: Sharkey)

3.1.5.3 The application shall integrate with Codeshare for real-time code collaboration between users.
(O: Fields, M1: Myers)

3.1.5.4 The application shall utilize third party API Google Drive for document sharing between users.
(O: Fields)

3.1.5.5 The application shall integrate with Git to share repositories between users.
(O: Fields)

3.2 Performance Requirements

3.2.1 The application shall retrieve entity objects first from entity Manager L2 cache.
(O: Crotzer)

3.2.2 The application shall be compatible with these versions of web browsers:

- Google Chrome Version: 72
- Safari Version: 12
- Microsoft Edge Version: 17

(O: Sharkey)

3.3 Assumptions and Constraints

N/A

3.4 Non-Functional Requirements

3.4.1 The application shall be written in Java.
(O: Sharkey)

3.4.2 The application shall utilize Java Server Faces and Primefaces for web application development.
(O: Sharkey)

3.4.3 The application shall utilize the Model–View–Controller design pattern and the facade design pattern.
3.4.4 The application shall employ entity classes for storing objects in the database. 
(O: Myers)

3.4.5 The application shall store users’ study preferences information in vectors. 
(O: Sharkey)

3.4.6 The application shall store users’ personal information. 
(O: Williams)

3.4.7 The application shall utilize Microsoft SQL Server Management Studio as the database management system. 
(O: Sharkey)

3.4.8 The application shall only display a buddy’s username by default, rather than their real name. 
(O: Sharkey, M1: Barsell)