COMP 3059 – Capstone Project I

Software Requirements Analysis and Design Assignment

1.0 Introduction

1.1 Purpose

The purpose of this document is to collect, analyse, and define the general needs and features of the Group Maker software. It outlines the specific problems the software is attempting to solve and includes a high-level description of the methods for solving these problems.

1.2 Scope

This document applies to the Group Maker software which will be developed by our capstone project group. The Group Maker software encompasses the database and database schema to store application information, the configuration of a server to support and host the software itself, and features of the software such as ranking algorithms, UI features, and both the online and mobile implementations.

2.0 System Overview

2.1 Project Perspective

Group Maker will be a self-contained software system. The project idea behind Group Maker software is to fill an existing gap in the schooling niche and it has no affiliation with existing systems, so it won't be replacing or following-on any software currently in the market.

2.2 System Context

Group Maker is designed to resolve the issues involving group projects, more specifically, the difficulty a student may have to find a group that matches with their study philosophy. The application will be available for

students, professors, and administrators to access their account and associated data via internet browser.

2.3 General Constraints

- Must be designed in such a way that the app is customizable for each college that might use it. This issue is due to making sure the application meets each college's needs (e.g. OAuth has to be updated for each specific college to authenticate the students at that college, admins must customize which features will be enabled, among others). This constraint will mostly impact the software implementation.
- Despite enabling Group Maker to fetch the most recent courses list for a program, the course information itself may become outdated after a couple of semesters so we are still constrained by professors' willingness to update their course information on our website. This constraint will mostly impact the software specification.
- We are constrained by the amount/type of information we can get from GBC. For once, we will not be able to get students grades or schedules without requiring the colleges permission (which may take a long time to acquire - not to mention the system security would have to be revised). This constraint will mostly impact the software design.
- Professor grading is another constraint we can only overcome with the college’s collaboration. The constraint is mainly due to Group Maker software not having access to the college’s database. As of now, we’re designing the software to not have direct grading and it’ll be only used to assist on grading (e.g. Professors can download lists of groups and group members and their perspective IDs). This constraint will mostly impact the software design.

2.4 Assumptions and Dependencies

Assumptions:

- Users will have access to a computer or mobile device with internet connection.
- Users will want to create their own groups outside of a specific course project with their own decided focus.
- Server support will be straightforward and relatively cheap to obtain.

Dependencies:

- Active involvement of administrators to add Courses and Students to the application's database
- Active involvement of professors to add projects and group number limitations within a given course so they are available for students.
- Active involvement of students to sign up, make profiles and make groups, request to join groups, accept auto-generated groups.
- Web server and database to store and deliver data to users.

3.0 Functional Requirements

A. List group member suggestions

a. Introduction
   i. A User who just created a group can see a list of users pulled from a pool of users filtered by given criteria - e.g. same campus, same department, same program, same semester, etc.
   ii. This list is sorted using the ranking algorithm
   iii. Group owner can send a join request to any given user that's displayed on the list

b. Inputs
   i. Group meta data
   ii. User meta data

c. Processing
   i. Sorting is done by an algorithm to display the list in order by likely match quality

d. Outputs

i. Sorted list of available users

B. List group suggestions

a. Introduction
   i. A User can seek to join a group from a pool of groups with available slots
   ii. Pool is comprised of users filtered by given criteria - e.g. same campus, department, program, or semester, etc. as user
   iii. This is basically the mirror image of feature 2

b. Inputs
   i. User meta data
   ii. Group meta data

c. Processing
   i. Sorting is done by an algorithm to display the list in order by likely match quality

d. Outputs
   i. Sorted list of available groups

C. Algorithms for sorting list of available users

a. Introduction
   i. A feature that attempts to improve the quality of each group by assessing the group attributes vs what each potential member offers based on their profile and their meta data.
   ii. A feature that attempts to sort the list of available members in such a way that the best candidates for the group are placed at the top
   iii. This feature would be also used in the auto join functionality
   iv. There would be a different algorithm for official group projects vs unofficial (i.e. groups that are not formed for the specific purpose of completing a school assignment)

b. Inputs
   i. Information about group
   ii. Information about user

c. Processing
   i. In the initial implementation, we expect the processing to be rudimentary, but other features can be added over time. Below is the bare minimum

ii. For official groups, simply rank members based on their group feedback score.
iii. For unofficial groups, rank based on the number of interests the users have that are compatible with the group focus.

d. Outputs
    i. Ranked list of suggested groups/users

D. Algorithms for sorting list of available groups

a. Introduction
    i. A feature that attempts to soft the list of available groups in such a way that the best ones for the user to join will appear at the top of the list
    ii. This feature would be used in the auto join functionality
    iii. There would be a different algorithm for official group projects vs unofficial

b. Inputs
    i. Information about group
    ii. Information about user

c. Processing
    i. Rank groups based on their cumulative score (averaged score of all users)

d. Outputs
    i. Ranked list of suggested groups/users

E. Admin management

a. Introduction
    i. This would be a dashboard implementation that displays reports about the usage of the app at a given school

b. Inputs
    i. Logging data that tracks some user activities such as groups created, groups closed, profile updates, ratings given, and other data meant to measure user engagement

c. Processing
    i. Compile the information, do any required transformation and historical comparisons to track trends

d. Output
    i. Business intelligence regarding app effectiveness

F. Create and edit user profile

a. Introduction
   i. Allow the user to create and edit their profile
b. Inputs
   i. Information entered by user
   ii. Existing information about that user from the DB (if any)
c. Processing
   i. Validate inputs, throw instructive errors if any
   ii. Enter into database

G. QR recognition for in-person group joining

a. Introduction
   i. Feature to make it quick for students to exchange contact details and join each other's group without going through the list of suggestions
b. Inputs
   i. Group ID
   ii. User ID
c. Processing
   i. Get group and add User ID to that group's member list
d. Outputs
   i. Success Boolean

H. Rating system to collect user feedback

a. Introduction
   i. Method to allow users to rate each other's group performance
b. Inputs
   i. Group ID, user ID of rater and ratee
c. Processing
   i. Create new DB entry in ratings for ratee, record that rater for given group has submitted the rating for the given group member (avoid double rating)
d. Outputs
   i. Success Boolean

I. Add users to verified user pools for official group assignments

a. Introduction

i. Used to make sure that lists of available users are all eligible to participate in the group project by verifying that they are actually enrolled in the same course

b. Inputs

i. Either a class email list provided by the professor user, or a list of user IDs provided by in class QR code scan

c. Processing

i. Associate the user IDs with the official project ID

d. Output

i. Success Boolean

J. User profile data entry

a. Introduction

i. User should be able to enter information about themselves in addition to the basic information required by their profile. The types of information can be extended over time as the ranking algorithm becomes more sophisticated.

b. Inputs

i. Items such as: interests, skills, technologies, personality questionnaire results, preferred tools, preferred meeting times.

ii. Most of the inputs would come from dropdown menus of options or nested dropdowns in order to avoid bad/unusable entries

c. Processing

i. Validate, compile and store all inputs

d. Output

i. Success Boolean

K. Regular group creation wizard

a. Introduction

i. For non-official groups. Feature allows user to select the pool of users from which to pull members (group scope). College level, campus level, program level, semester, etc.

ii. Then group creator sets the focus of the group. E.g. what the group is about. Some predefined categories will be provided such as Gaming, Books, Movies, Music, Fitness, Cooking, Politics, Philosophy, Writing etc. There can be sub-categories to many of these. The categories will be used by the sorting suggestion

algorithm. And some professional categories can also be included such as project management

b. Inputs
    i. Group creator profile info
    ii. User inputs for group focus
c. Processing
    i. Validate, then add the group info the database
d. Output
    i. Success Boolean

L. Communication method auto suggestion

a. Introduction
    i. On user account creation, one aspect of profile to fill out is preferred communication platform. This feature would observe the preferred methods for all group members and automatically suggest the one which most users have
b. Inputs
    i. User profile info
c. Processing
    i. Compare all users, count the number of occurrences of each communication preference
d. Output
    i. Link to initiate group chat with group members using the most common platform

M. Available member designation to fill every group

a. Introduction
    i. Shows the pool of users that have not found groups so that people in groups can invite them to join
b. Inputs
    i. User's group availability status
    ii. User's ratings
c. Processing
    i. Determine and return available users
d. Output
    i. List of users ranked as most valuable for said group

N. Automatic profile creation for students who join through in-class QR code

a. Introduction
    i. Student should be able to join an official group project easily by taking a photo of the QR code. In the case that they do not have a profile already setup, this should then automatically fill in their profile with campus, program, semester and project info
b. Inputs
    i. Project ID
    ii. (new) user ID
c. Processing
    i. Check if user has account, if not create one, then associate student ID with project ID and populate student info RE college, campus, program, semester, section, course with appropriate info based on the project
    ii. Associate student ID with project ID in database
d. Output
    i. Return the appropriate list of available groups,

O. Web short links for easy group setup sans mobile app

a. Introduction
    i. Feature to make it quick for students to exchange contact details and join each other's group without going through the list of suggestions
    ii. Same as the QR code feature but using web links instead
b. Inputs
    i. Group ID
    ii. User ID
c. Processing
    i. Get group and add User ID to that group's member list
d. Outputs
    i. Success Boolean

P. O-Auth for profile connection/creation

a. Introduction
    i. Method of ensuring that user is a member of the correct institution
b. Inputs
    i. User college email and password
c. Processing

i. Verify email and password through o-auth service

d. Output

i. Success Boolean and session

3.2 Use Cases

A. List group member suggestions

a. Group leader ->

i. Check list of users ranked for their group

B. List group suggestions

a. User ->

i. Check available groups ranked to match them

C. Algorithms for sorting list of available users

a. System ->

i. Calculate which users fit which groups best

D. Algorithms for sorting list of available groups

a. System ->

i. Calculate which groups should be ranked above others

E. Admin management

a. Admin ->

i. CRUD programs

ii. CRUD groups

iii. CRUD users

F. Create and edit user profile

a. User ->

i. Create account

ii. Modify account

iii. Delete account

G. QR recognition for in-person group joining

a. User ->

i. Scan QR to join group lobby

ii. Scan QR to join group

b. System ->

i. Generate QR

H. Rating system to collect user feedback
    a. User ->
        i. Rate other members when project finishes
    b. System
        i. -> Update other user profiles
I. Verified user pools for official class group assignments
    a. Teacher ->
        i. Create group lobby limited to specific users
J. User profile data entry
    a. User ->
        i. Edit granular profile settings
    b. System ->
        i. Update account
K. Regular group creation wizard
    a. User ->
        i. Create group
        ii. Modify group
L. Communication method auto suggestion
    a. System ->
        i. Select most common user communication preferences from profile
M. Available member designation to fill every group
    a. System ->
        i. Ensure users are available for groups
N. Automatic profile creation for students who join through in-class QR code
    a. System ->
        i. Create blank profile for user when they join through QR code
O. Web short links for easy group setup sans mobile app
    a. System ->
        i. Generate web short link for group lobby
P. OAuth for profile connection/creation
    a. System ->
        i. Allow user to sign up through OAuth
    b. User ->
        i. OAuth themselves

3.3 Data Modelling and Analysis

- Normalized Data Model Diagram

  See attached file
- Activity Diagrams

  See attached files prefixed “AD”
- Sequence Diagrams

  See attached files prefixed “SD”
- UML Class Diagram

  See attached file

3.4 Process Modelling

- Data Flow Diagram

  See attached file

4.0 Non-Functional Requirements

The main non-functional requirements are speed and ease of use. Users should be able to quickly find their way into a group and one that is tailored to their needs without much effort. The first requirement of this is performance. As most of the group making process is formed at specific times as work is assigned, it is vital that the process can be completed in seconds. Our system will automatically match groups and people together without input being required from the user, then prompt the user that a group exists that would fit them well. Profile creation should also be able to be completed in a few seconds, but it will only be offered to users as a choice rather than forced upon them and taking extra time. There should be less than 30 seconds from install time until use of the system to prepare groups.

Being constantly available is another requirement as teachers may assign work at any time. If we cannot predict when the system will need to be used, it must always be available. For this reason, downtime should be limited to only short

periods of time. It should also be accessible from anywhere in the world, as long as an internet connection is available. The system should also work when different versions of the app are available so that users on different versions of the app can still communicate and join groups as normal. Target downtime is less than 30 minutes per month.

5.0 Logical Database Requirements

A database is necessary for our project. All required user and group data formats are basic and can be represented as strings and integers. Current data storage plans require that a database be accessible at any time from any location. For this reason, a cloud database management system is going to be used. Providers that have been explored are Amazon Web Services and Google Cloud. Both offer multiple options for data storage which will suit our needs. A large benefit of going using a cloud database manager is that the infrastructure does not need to be maintained by us, with the drawback that we're unable to manage them if there are any outages. Considering the scale of our requirements along with the infrastructure available through these companies, when addressing the risk of outages versus the risk of hosting our own servers, we have chosen to use an external cloud database provider.

Data retention

Data will be retained only if it is needed for future algorithmic group prediction. While this could balloon into a large amount of data, it is planned to consolidate any data from groups and ratings that have occurred in the past up to a point. Data within the last 6 months will be stored for use while old records will be deleted. All data retention policies will abide by any legislation as well, such as the GDPR.

Data integrity

Using a cloud database storage system assists in physical data integrity. Not long after some data is uploaded, it can be backed up and stored on multiple disks across the internet. This should be a large safeguard against data loss. Logical data integrity is an issue that is more pressing. While our data could be used to benefit and shape good user groups, bad actors could also attempt to disrupt group creation. One way to protect against this is to only accept positive ratings for people i.e., a user could only be asked to rate a person as prompt or skilled or give no rating at all. We will also attempt to store only small amounts of data necessary for the most basic features of a profile as there will be waves of data due to work being assigned to students at specific times of day. The database will be required to use appropriate foreign keys and not allow data duplication, as keeping our databases small will be required for speedy access during busy times.

6.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Group member	Alexander Balez		2020-11-14
Group member	Barrington Venables		2020-11-14
Group member	Kevin Ufkes		2020-11-14
Group member	Thiago Macielhissa		2020-11-14