

Concept of Operations

Brendan College, Alejandro Fernandez, Charles Karlson, Samantha Maddox

Project 1: Group Matching App

COP 4331, Fall 2021

Contents of this Document

The Current System

The Proposed System

- Motivation
- Users and Modes of Operation
- Operational Scenarios
- Operational Features
- Analysis

The Current System

Meetup is a similar application. It's a "platform for finding and building local communities. People use Meetup to meet new people, learn new things, find support, get out of their comfort zones, and pursue their passions, together" ([meetup.com/about](https://www.meetup.com/about)). Facebook is also an application accomplishing similar goals to myMeet. Facebook Groups are "a place for group communication, letting people share their common interests and express their opinions. Groups let people come together around a common cause, issue, or activity to organize, express objectives, discuss issues, post photos, and share related content" ([facebook.com/groups](https://www.facebook.com/groups)).

The Proposed System: Motivation

Users may want to use a simpler social media platform that does not collect or sell their personal data, trends, or likeness. myMeet is much simpler than other apps currently available and will not reveal user information to third parties. Simpler software may be easier to use for those who are unfamiliar with how to use current social media. By storing less information than other applications, myMeet may also be less expensive to store and maintain. Other applications like Facebook and Tinder may not be within the scope of those who look to use myMeet.

The Proposed System: Users and Modes of Operation

For most operations, there will only be one class of user. Users will have the following fields: username, password, first name, last name, security question, security answer, groups, interests, friends, profile picture, as well as a message board. Within groups, some users (group administrators) will have elevated privileges to manage their group and group's events.

myMeet will contain three main modes of operation: users, groups, and events. In users, individual users will be able to connect with other users that have similar interests. Users will be able to direct message other users they connect with, as well as "friend" those other users. In groups, users will be able to either create or find and join groups that revolve around their interests. Users will be able to send messages to a group text chat and create or join group events. In events, users will be able to join events centered around their interests, create individual events, and see uploaded pictures from events they have attended.

The Proposed System: Operational Scenarios

The primary use for the user mode of operations is to connect with other users. For example, myMeet may suggest a list of profiles of users that have similar interests to one's own interests. One would then be able to view those profiles, and message the user and/or send a friend request to that user.

The goal of group operations is for users to find and join groups, and then be able to message other users in the group, as well as create events. Group operations would contain some aspects of user and event operations. For example, like with user operations, myMeet may suggest a list of groups that have similar interests to one's own interests. One would then be able to view the group and join it and send messages to the group's chatroom. One could also view individual profiles of group members and perform user operations. Events would be able to be created, joined, and viewed like in event operations. Alternatively, one could create their own group, and administrate it themselves or promote group users to administrate it. Administration would include: editing or deleting events, removing or promoting users, and changing the group name and picture.

In event operations, users should be able to find and join events, as well as view event information and pictures from those events if they have joined them. For example, one can find an event before it begins and register for it. They then can view information and pictures for that event. Alternatively, one can create an event and provide the name, location, time, and date for it, and allow users to join it.

Predatory users need to be considered when building social media platforms. All users will have access to a block function that will block and mute users the function is used on. In addition, group and event administrators will have the ability to review and remove pictures flagged by users. In any case, malicious action has the ability to be arrested by users, whether it be by blocking malicious users, removing malicious users from groups and events, and removing inappropriate pictures from event pages.

The Proposed System: Operational Features

Must Have:

- Users
 - Create, manage, delete

- Profile: interests
 - Message
- Groups
 - Create, manage, delete
 - Join
 - Administrate (kick)
- Events
 - Create, manage, delete
 - Join
 - Host pictures
- Database

Would Like to Have:

- Users
 - Block/mute
 - Profile: picture, groups, friends
- Groups
 - Chatroom
 - Interests
- Events
 - Flag pictures

The Proposed System: Analysis

myMeet will be written in Java, most likely using Eclipse and Maven to manage it. GitHub will be used to host the project repository. The app will be developed as a Java app, and hosted on a web server. JavaScript will be used intermittently on the hosted web application. MySQL may be used for database management of users, groups, and events.

Learning will be needed to become familiar with Maven, Git, Swing, and MySQL. One limitation of hosting a Java app is that we will not be able to have hyperlinks for individual users, groups, or events. An internet connection will be needed to connect to the database. myMeet is not intended for use in hostile or volatile environments, or where meeting others may result in injury or death. myMeet is not intended for use by those under 18.

Java is a versatile language, and most of our goals may be accomplished using Java; our only limitation is the skills of the developers. Unfortunately, Java will consume a lot of memory. C++ and Python may have also been used. C++ is a more efficient language, and Python is great for web interaction, but Java is the chosen language of the developers. Limitations will be present in the web hosting the application, as it would not be natively built as a website. The current selection is the best possible, as the developers can accomplish more in the given time frame in Java than if they were to spend time learning and applying other methods.

Project Management Plan

Brendan College, Alejandro Fernandez, Charles Karlson, Samantha Maddox

Project 1: Group Matching App

COP 4331, Fall 2021

Contents of this Document

Project Overview

Applicable Standards

Deliverables

Software Life Cycle Process

Tools and Computing Environment

Configuration Management

Quality Assurance

Risk Management

Table of Work Packages, Time Estimates, and Assignments

GANNT or PERT Chart

Technical Progress Metrics

Plan for tracking, control, and reporting of progress

Project Overview: myMeet is an interest-matching social media application. Users are able to match with other users based on their interests, as well as message and friend those other users. Users can also create and join groups centered around their interests, and message with other users in their group's chatroom. Users or groups can also create and join events, and see pictures from events they have attended.

Applicable Standards

- Standard Java naming conventions (<https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html>)
- Header comments describing each method
- Maximum 1 GB for any single file
- The above may change as the realities of programming become more apparent

Deliverables

Artifact	Due Dates
Individual Weekly Progress Reports	Weekly (Fridays) submission throughout the semester through webcourses
Concept of Operations	9/17/21
Software Project Management Plan (SPMP)	9/17/21
Software Requirements Specification (SRS)	9/17/21
High-Level Design	9/24/21
Detailed Design	10/1/21
Test Plan	10/15/21
Test Results	10/22/21
Source, Executable, Build Instructions	12/1/21

Software Life Cycle Process: The Agile software model will be followed due to the relative small size of the team, and their inexperience compared to industry expectations. Working code will be preferred over detailed documentation, and frequent meetings will be needed to ensure integrity of the project.

Tools and Computing Environment: Programming will occur in Java 11 (LTS) on Windows 10 utilizing Eclipse. It will be compiled with OpenJDK 11.0.11. It will be assumed that any libraries offered by Oracle may be used. Swing will be used for GUI elements. More libraries may be used as the project progresses.

Configuration Management: Git will be used for version and change control, with the project being hosted in a GitHub repository. Each developer will be responsible for a certain set of class documents.

Quality Assurance: Each developer will be responsible for ensuring programmed methods and classes are consistent and functional. The QA manager will ensure all classes work in conjunction with the database. Results will be reported to the group as a whole. Project and QA managers are responsible for ensuring compliance.

Risk Management: Time management will be a key factor due to the programmers' inexperience. Nobody must be able to remove database entries they are not authorized to, and myMeet will be programmed to not allow such action. Safeguards will be put in to arrest predatory behavior.

Table of Work Packages, Time Estimates, and Assignments: It is unknown to the developers how such work packages will be distributed amongst the team. This section will be updated at a later date.

Technical Progress Metrics: Modules will be worked on until the module produces the expected output. Developers will distribute work amongst themselves and verify work with the project manager. It will be determined by all members of the group if progress is being made at specific steps.

Plan for tracking, control, and reporting of progress

Weekly group meetings will be conducted on Fridays detailing the progress made that week, and which tasks are to be done the next week. GitHub will be used for documenting and tracking completeness of modules. Discord and SMS will be used for communicating issues and concerns. Progress reports will be worked on during the weekly sessions and posted to Webcourses.

Software Requirements Specification

Brendan College, Alejandro Fernandez, Charles Karlson, Samantha Maddox

Project 1: Group Matching App

COP 4331, Fall 2021

Contents of this Document

Introduction

- Software to be Produced
- Applicable Standards

Definition, Acronyms, and Abbreviations

Product Overview

- Assumptions
- Stakeholders
- Event Table
- Use Case Diagram
- Use Case Descriptions

Specific Requirements

- Functional Requirements
- Interface Requirements
- Physical Environment Requirements
- Users and Human Factors Requirements
- Documentation Requirements
- Data Requirements
- Resource Requirements
- Security Requirements
- Quality Assurance Requirements

Supporting Material

Section 1: Introduction

Software to be Produced:

- myMeet will be a web-hosted Java application with an attached MySQL database. It will allow users to connect with other users, create and join groups, as well as create and join events.

Applicable Standards

- It will exclusively run on a web browser
- All other standards used are in the Project Management Plan

Definitions, Acronyms, and Abbreviations

- None

Section 2: Product Overview

Assumptions:

- Users are 18 or older
- All modules (database, website, and application) are compatible

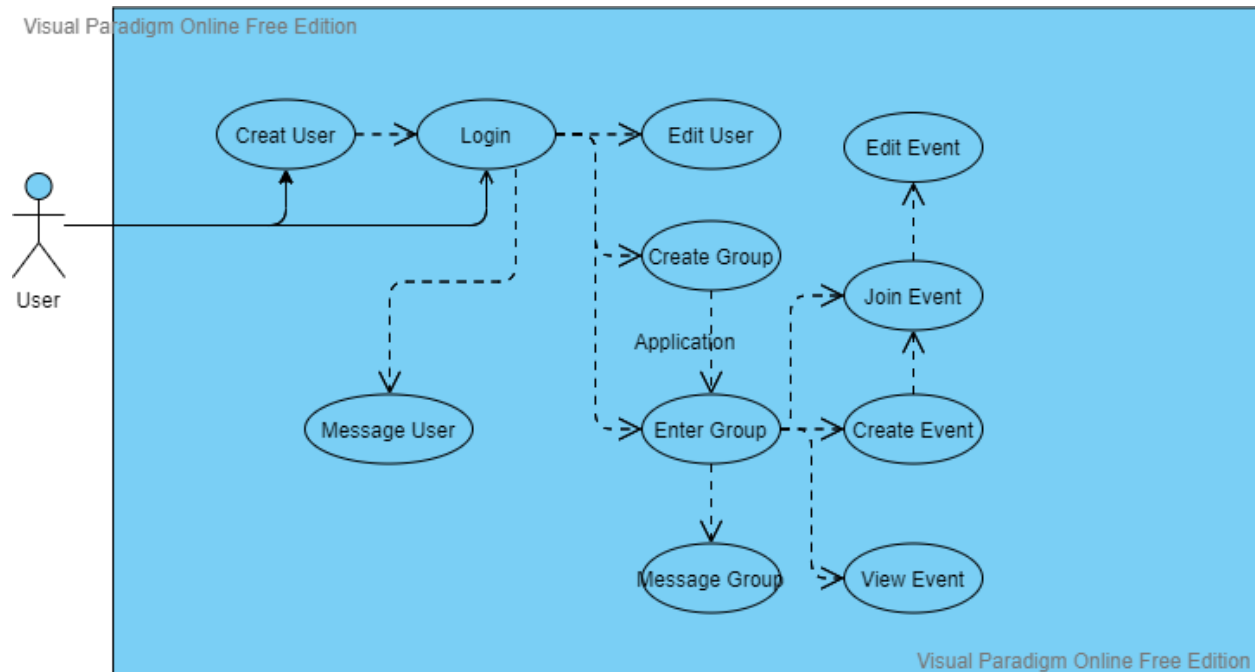
Stakeholders:

- Users: will be using the application
- Programming team: will need to have a functioning program to receive a grade
- Professors/TAs: will receive the program and provide grades to the programming team

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Create user	User creates profile	Confirmation of creation	Add to table
Delete user	User deletes user	Confirmation of deletion	Drop from table
Create group	User create group	Confirmation of creation	Add to table
Delete group	User deletes group	Confirmation of deletion	Drop from table
Create event	User creates event	Confirmation of creation	Add to table
Delete event	User deletes event	Confirmation of deletion	Drop from table

Use Case Diagram



Use Case Descriptions:

- Create user
- Login
 - Once users log in, they have access to all normal functions of the program

Section 3: Specific Requirements

No: 1
Statement: Users shall be able to manage profiles
Source: Basic functionality
Dependency: User must have a valid account or be able to create one
Conflicts: None
Supporting Materials: Above
Evaluation Method: User is able to influence the database
Revision History: n/a

No: 2
Statement: Users shall be able to manage groups
Source: Basic functionality
Dependency: User must have a valid account
Conflicts: None
Supporting Materials: Above
Evaluation Method: User is able to influence the database
Revision History: n/a

No: 3
Statement: Users shall be able to manage events
Source: Basic functionality
Dependency: User must have a valid account and be a part of a group
Conflicts: None
Supporting Materials: Above
Evaluation Method: User is able to influence the database
Revision History: n/a

3.1 Functional Requirements

- Sanitize text inputs, reject bad words
- Arrest spam attempts
- Allow users to create, manage, and delete profiles, groups, and events

3.2 Interface Requirements

- Read and write to/from the database via the website (timing/sync issues not present)

- Data will be accurate to the latest refresh
- Information that comes as a result of user interaction with the application will accordingly be updated in the database

3.3 Physical Environment Requirements

- Java application will be hosted on a website, which will be hosted on a third-party hosting service
- Database will be hosted on Eustis

3.4 User and Human Factors Requirements

- System will support one class of user (substate for administration of groups)
- Simple users, assumed average skill level
- Simple how-to page on how to use myMeet
- Accessibility options will be through the user's own browser and OS
- System will detect and arrest spamming attempts (e.g. creating multiple users, groups, events)
- System will only allow registration of those at or above the appropriate age

3.5 Documentation Requirements

- Simple online page on how to use myMeet
 - Users assumed to be an average social media user (any age)
 - Web page (or PDF on web page) hosted to show normal functions of the application
 - How to interact with other users
 - How to interact with groups
 - How to create and administrate groups
 - How to interact with events
 - How to create and administrate events

3.6 Data Requirements

- Interest matching arithmetic will need to be defined at some point

3.7 Resource Requirements

- Team of 4 developers to build and maintain
- Communal spaces on the UCF campus will be used for physical meetings
- Discord will be used for virtual communication
- No monetary funding
- Open-source software
- Completed by end of semester, fall 2021

3.8 Security Requirements

- Database must not be accessible by users or third parties
 - Only official requests (e.g. by the program or by developers)

- Sensitive information (e.g. passwords) will be hashed
- Users will be isolated from other users, only able to make requests to the program for the program to read/write the database
- Users will have security questions and answers for resetting a password
- Daily backups stored onsite
- Weekly backups stored offsite

3.9 Quality Assurance Requirements

- Functionality above all else
- Concurrent same user logons will kick the oldest user logged in
- Hard user timeouts
- Weekly downtime for maintenance
- Reliability and security will be project goals

Section 4: Supporting Material

- None