Detailed Design

Brendan College, Alejandro Fernandez, Charles Karlson, Samantha Maddox (Group 6)

Project 1: Group Matching App

COP 4331, Fall 2021

Contents of this Document

Detailed Design Issues

Major Design Issues

Reliability:

- Will be a concern due to time and testing constraints

Reliability will be a concern due to our decision to have a web-based application. With that comes the risk of issues between internet connectivity, server complications, and possible long load times. In addition, our decision to host the website on Eustis could possibly result in website downtimes due to the server not being available because of maintenance or heavy traffic. We chose Eustis due to it providing a free server to host our application and because all the developers have experience using Eustis.

Reusability:

- Continuous instance of myMeet unless reset or shut down by developers
- Service can easily be replicated and rebranded if source code present

Our code will have components with high cohesion, low coupling, and a well-defined interface. To query our database, we will have a reusable code to save time and have consistency throughout our classes. We will implement a syntax and style throughout our code to ensure readability and improve the code structure. We can replicate the service and rebrand if necessary due to our high level of generic and reusable code.

Maintainability:

- Under constant watch by developers
- Bug reports should be

After strenuous testing, and with the help of bug reports, we should be able to maintain the integrity of the application while ensuring everything backend is running as it should be. Internally, we will have exceptions and flags that will catch any faults preventing any damage to the user experience. We will implement a syntax and style throughout our code to ensure readability and improve the code structure.

Testability:

- Developers will evaluate use cases and will perform tests based on those use cases
- Will ensure performance is consistent

Developers will have thorough testcases that test the modules and website. We will feed select inputs through the modules and applications and match it with a correct selected output. After performing unit tests, developers will determine if the product is good enough for a general user to use.

Performance:

- Resource-light for users
- Speed not imperative for operations performed in the application
- Lightweight database query

Developers expect relatively quick response times between the website and the server. We are using JavaScript due to how lightweight and response it is. We are balancing the workload between the servers and JavaScript present on the website to ensure good response times.

Portability:

- Users
    - Extremely portable
    - Reliant only on an internet connection and browser
    - Resource-light
- Developers
    - Unportable
    - One instance of myMeet at a time
    - One database
    - Setup is involved (but not necessarily intensive)

Since this is a web-based application, we are only making it ideally suited for online use. Developers chose this option because they had more experience versus offline applications. Since we are using an online application most modern browsers should be able to interact with the website. Developers are not planning to port the application to iOS or Android and are exclusively testing on Windows 10 systems. Developers are using JavaScript, html, and css elements to be compatible with most browsers so users from different environments can use the application.

Security:

- Database security
    - Database restrictions encoded in Java program
    - Duplication of primary keys disallowed
- Passwords and sensitive information hashed
    - Password reset functionality
- Focus on user privacy
- Eventually it will become an open source

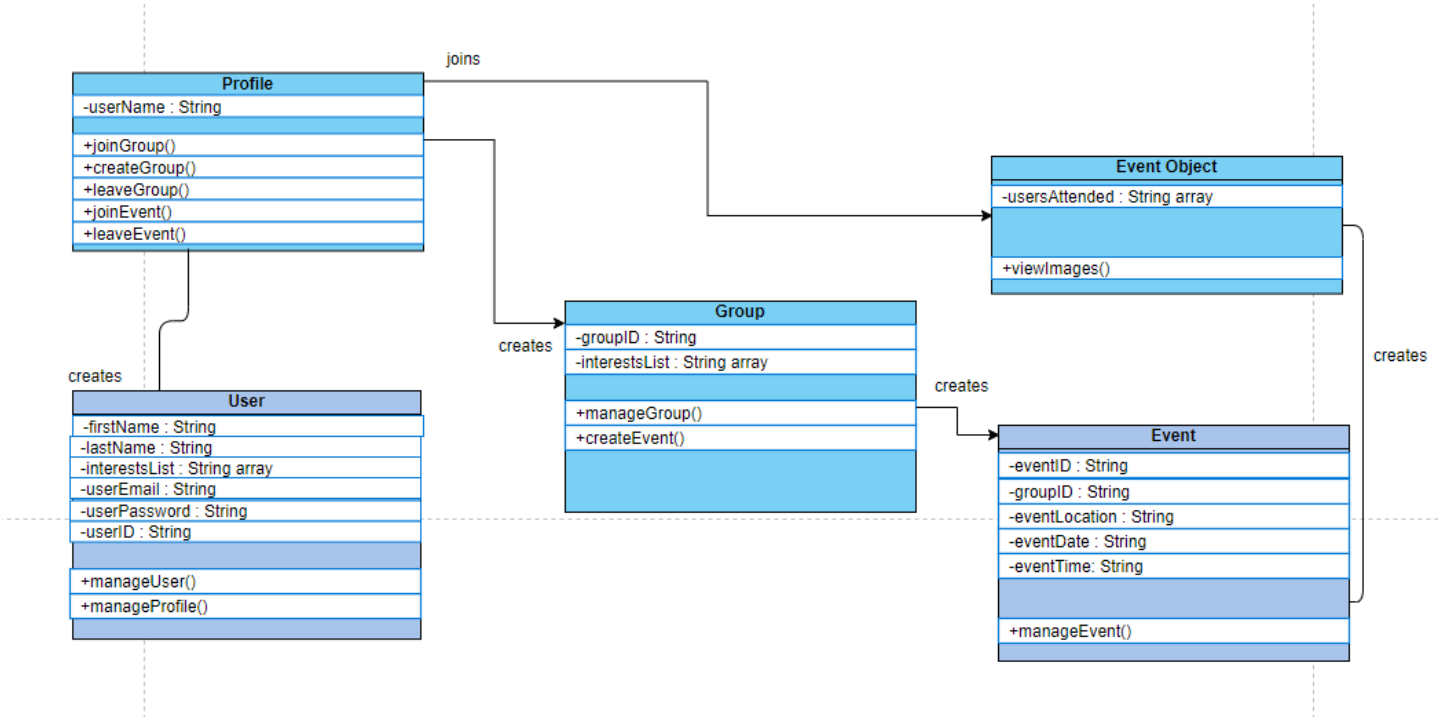
User's private information will be closed. No other parties except for the developers will have access to it. Sensitive fields will be hashed and essentially encrypted to keep user's information safe. Classes will be encapsulated. If users forget their password, we have a password recovery system in place that sends a password recovery link to their email.

Safety:

- Safeguards against malicious behavior
    - Kick users
    - Block users
    - Minimum age
- User privacy

Group administrators will have the ability to kick and block users that they deem harmful in their own judgement. In addition, group administrators will have the ability to delete harmful images posted to those who attend the events at their discretion. With our security implementations in place, users will only have access to each other's personal information if they choose to share that with others; keeping our user's safe. Users can also change their personal information if sensitive information is compromised.

Detailed Design Information



Trace of Requirements to Design

Requirements from SRS document:

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

The detailed design diagram shows that users can manage their profile. The modules User and Profile satisfy Requirements No. 1.

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

The detailed design diagram shows that users can manage their groups. The module Group satisfies Requirements No. 2.

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

The detailed design diagram shows that users can manage their events. The modules Event and Event Object satisfy Requirements No. 3.