**Software Requirements and Design Document**

**For**

**Group 5**

Version 2.0

**Authors**:

Benjamin Zech

Corin Bradley

Gregory Garmen

Ian Carter

Wesley Harris

# Overview (5 points)

*Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).*

Our app is a way for FSU students and local Tallahassee businesses to create and post social events in which others can attend. Each created event will have members who can post messages to a feed so that others can see. This app is a way of organizing events while creating a fun and social atmosphere.

# Functional Requirements (10 points)

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. **Student Account Creation** *→* ***High Priority*** *-* This allows students to sign up for the app. Student account creation is needed to use the app.
2. **Business Account Creation** *→* ***High******Priority*** *-* This allows businesses to sign up for the app. Business account creation is needed to use the app.
3. **Account Login** *→* ***High Priority*** *-* This allows users to login to the app. User login is needed to use the app.
4. **View Profile -> Medium Priority** - This allows users to view their created profiles and make edits to their profile pictures. This page will be viewable by other app users.
5. **Profile Search → High Priority** - This allows users to search for other user profiles using a search box
6. **Direct Messaging → Medium Priority** - This allows users to communicate with one another one on one within the app.

# Non-functional Requirements (10 points)

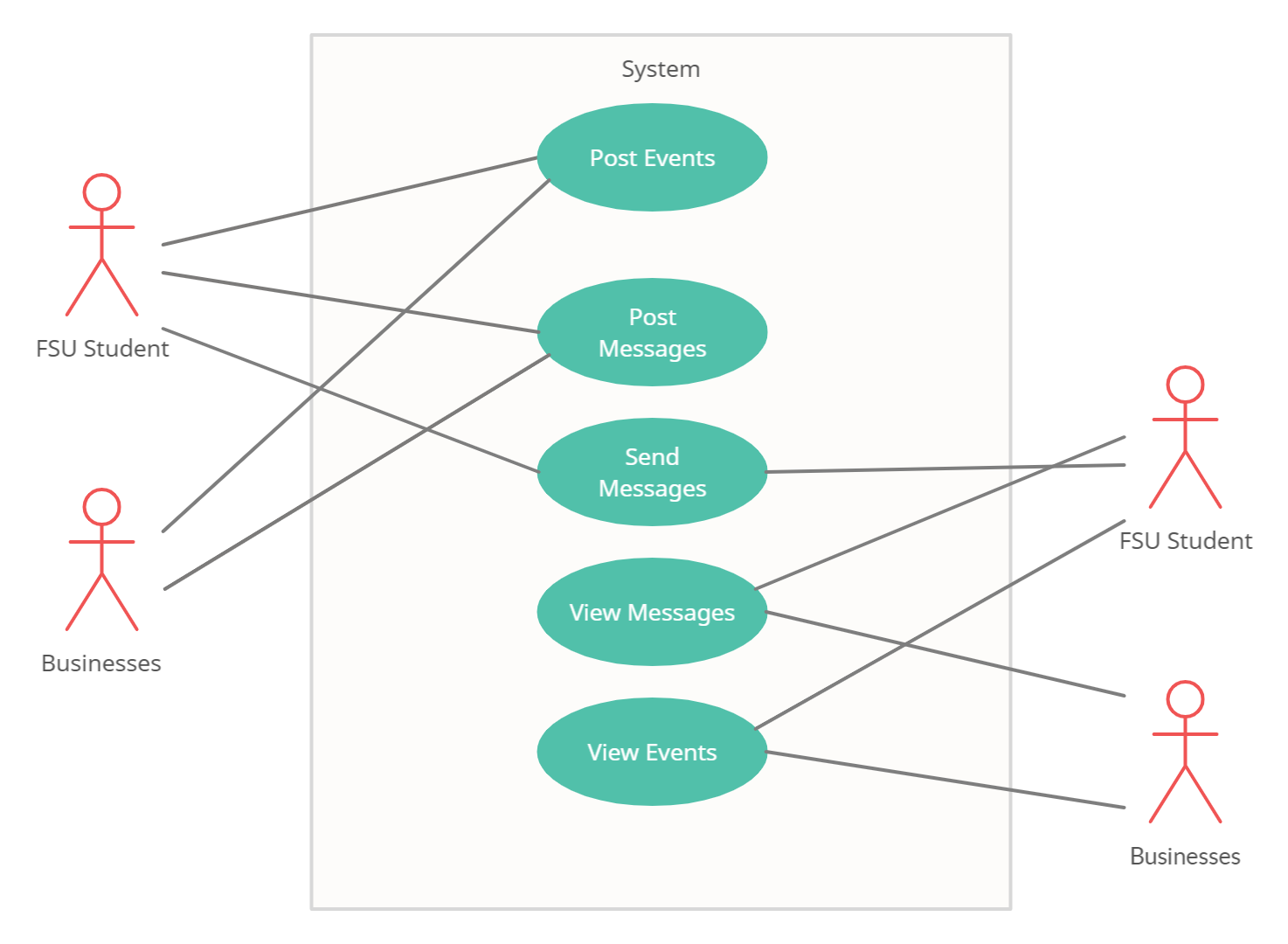
*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

1. **App running on different Android Phones**
2. **FSU only account creation** - Making sure that only FSU email addresses can sign up for an account

# Use Case Diagram (10 points)

*This section presents the* ***use case diagram*** *and the* ***textual descriptions*** *of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

***Textual descriptions of use cases****: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*



# Class Diagram and/or Sequence Diagrams (15 points)

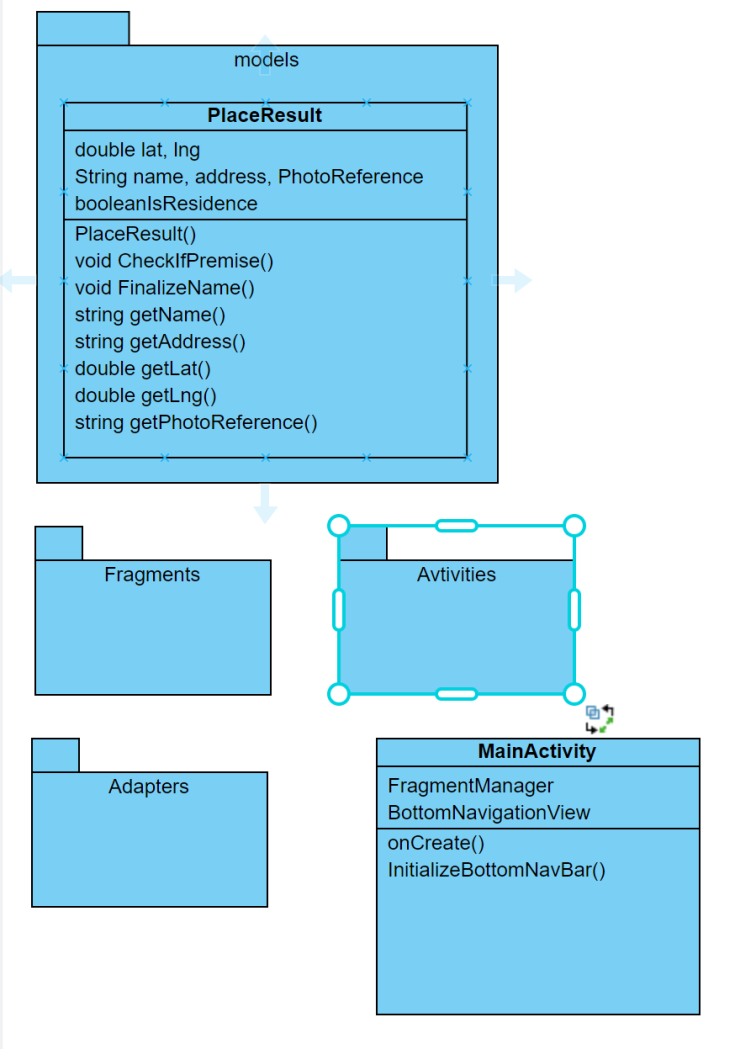
*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

**

# Operating Environment (5 points)

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

The app will function only on Android mobile devices running API level 28 or above. The app will need to be run from the Android Studio IDE.

# Assumptions and Dependencies (5 points)

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

An assumption we will have to make, for now, is that users signing up are indeed either FSU students or a Tallahassee business. We have a very loose guideline for allowing users to sign up, and as students, we do not have the necessary resources to enforce or confirm truthful users. In a real app, we would need to confirm that an FSU email is legitimate and that someone claiming to be a business is actually associated with the business. So for the sake of developing this app, we will assume truthful user signup.