Micoscu

SE 305 – Software Specification and Design

Term Project Design Document

Ahmet Muazim Tunç, Mirkan Çalışkan, Çağlar Bilgin

December 22, 2017

1. Introduction

Micoscu designed to match people with each other who miss the excitement, sharing of the real experience. It aims to help players to find partners to play any games or do any sport, activities.

Activities widen to street soccers, outdoor hikes, cycling to board games which can be played in cafes, restaurants or any facilities those are willing to participate and increase their profit.

With Micoscu, our business partners can host our users in various events and games, or they can create their own events and let the system notify nearby users. By the way they can enjoy the reputation and extra clients that provided by Micoscu, or they can simply let us advertise their menus and promotion.

2. Problem Definition

We want a world of people to have fun by socializing meanwhile not having any darkside of technology, where everyone is trying to reprogram their sense of entertainment and finds opponents or participant with their smartphone

Humans need to have fun, but having fun behind the computer or smartphone screens and staying inactive comes with various side effects in our health. Mostly the young generation have those side effects. Screen time has effect on our sleep quality, having more online friends than the real ones has impact on our both social and professional life, sitting rather than being active has impact on our body.

3. Proposed System Design

3.1. Requirements

**Functional Requirements**

1) Users shall be able to send friend request

2) Users shall be able to create a match

3) Users shall be able to determine date and location

4) Event results shall be added to systems

5) Users shall be able to view results

6) Users shall be able to comment, like and share event results

7) Users shall be able to login

**Non-functional Requirements**

1) Users shall be able to see friends suggestions based by location and hobbies

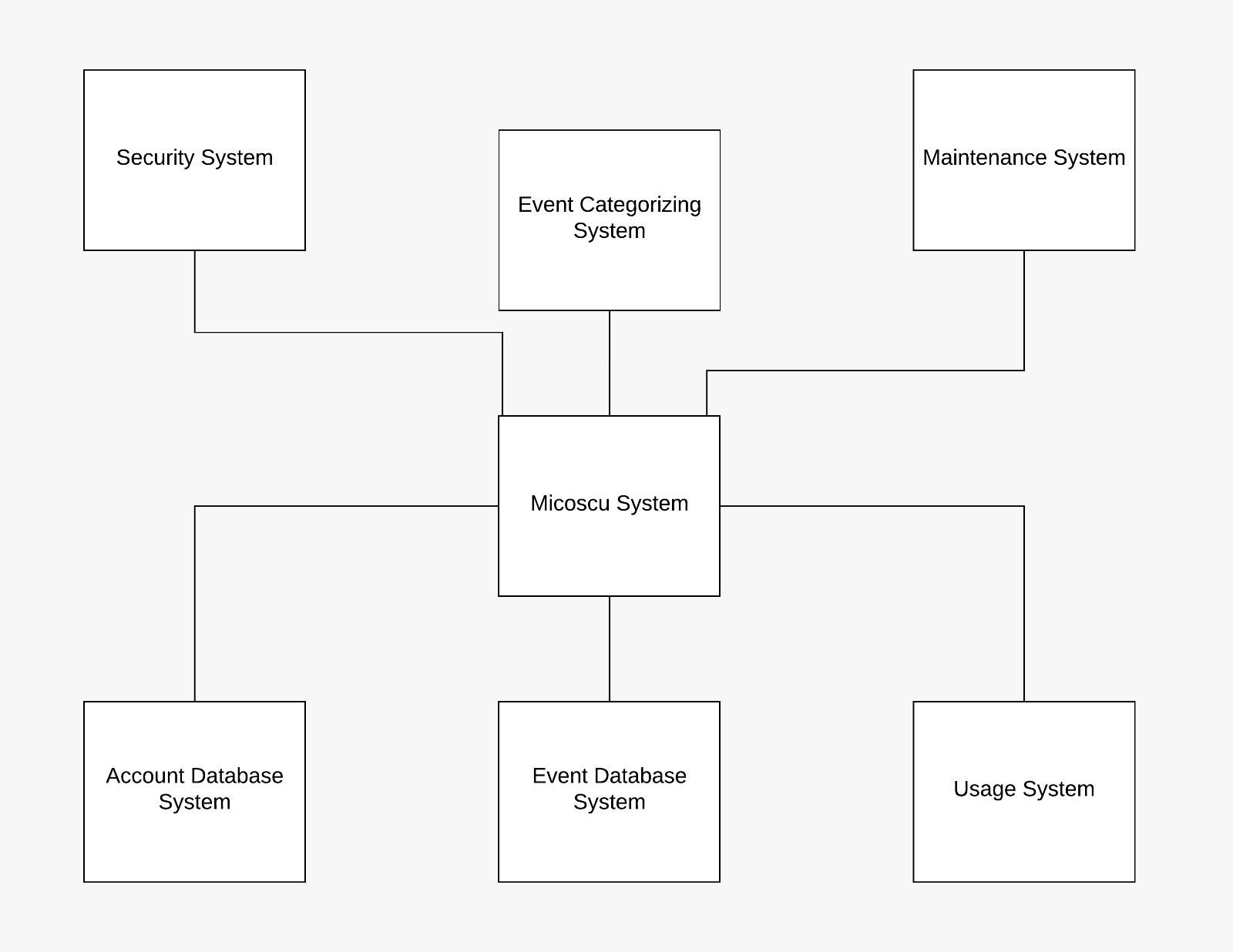
2) Users shall be able to agree on date and location by handshake method

3) Event results shall be added to systems only by authorized personel

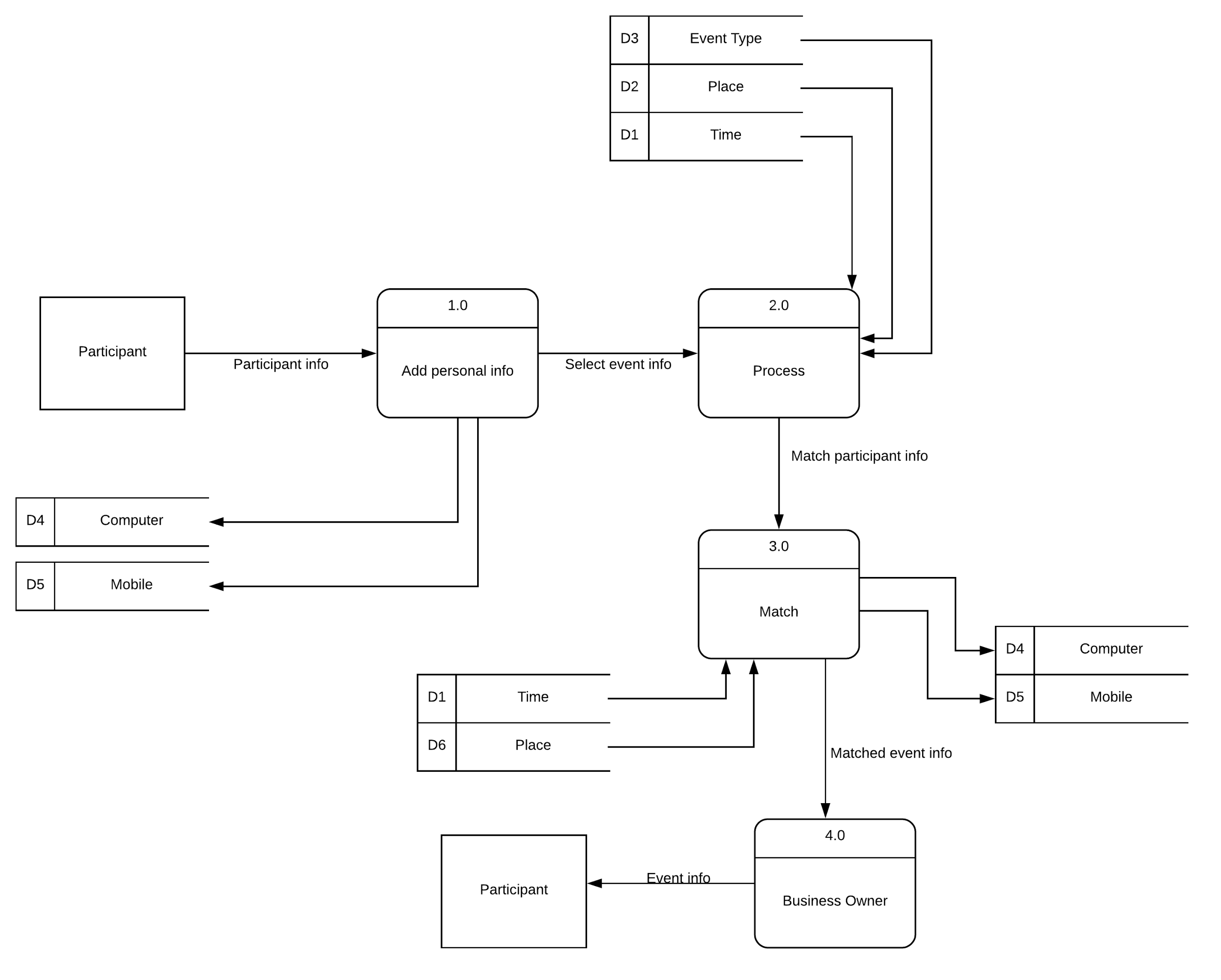
4) Users comments shall be sorted by upvotes and downvotes

5) Users shall be able to login by their social media accounts

3.2. Context Model

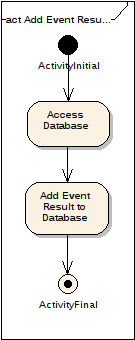


3.3. Data Flow Diagram

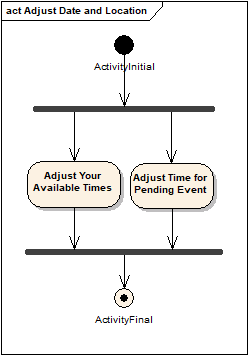


3.4. Activity Diagrams

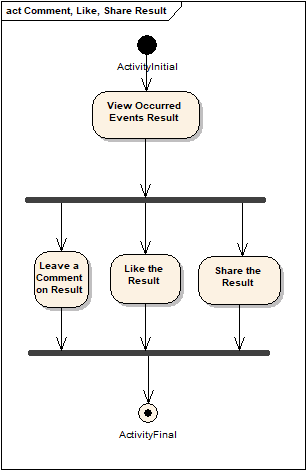
3.4.1 - Add Event Result on System - Activity Diagram 1:



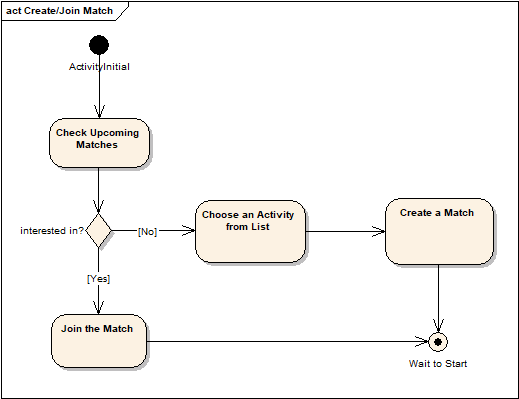
3.4.2 - Adjust Date and Location - Activity Diagram 2:



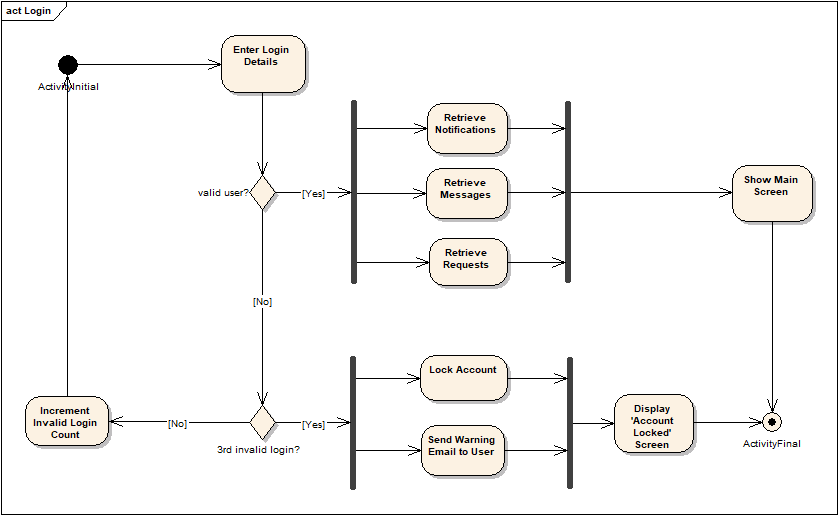
3.4.3 - Comment, Like Share Result - Activity Diagram 3:



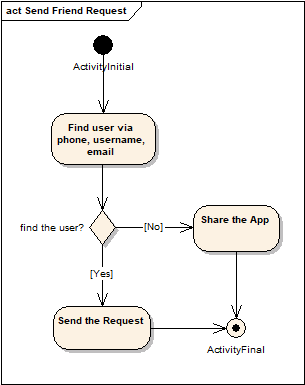
3.4.4 - Create/Join Match - Activity Diagram 4:



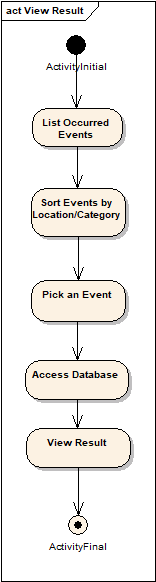
3.4.5 - Login - Activity Diagram 5:



3.4.6 - Send Friend Request - Activity Diagram 6:

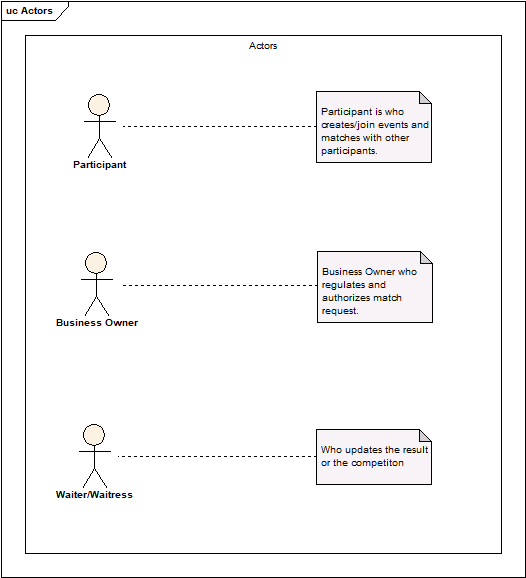


3.4.7 - Adjust Date and Location - Activity Diagram 7:

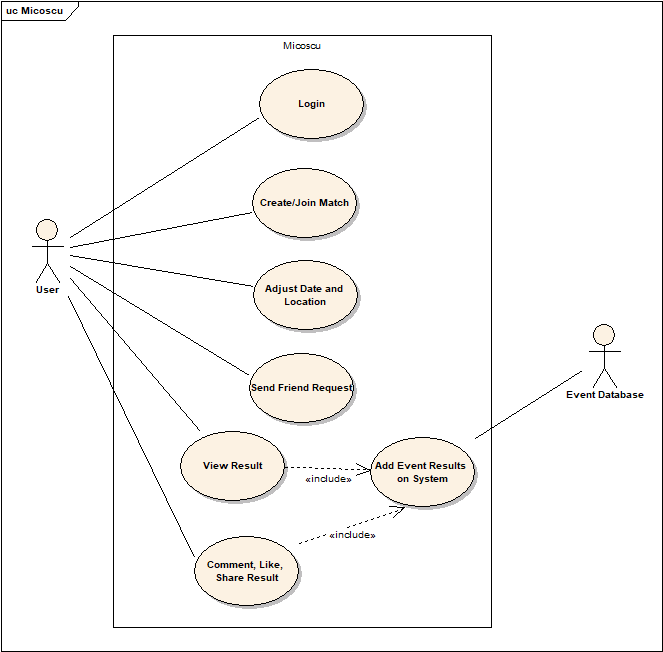


3.5. Use Cases

3.5.1. Actors



3.5.2. Use Case Diagrams



3.6. Use Case Text

1. User logins to system.

if user doesn't have an account, first sign up, then login

2. User checks messages, requests or sends friend requests.

3. User joins one of current events

if user doesn't interested in any event, user creates an event.

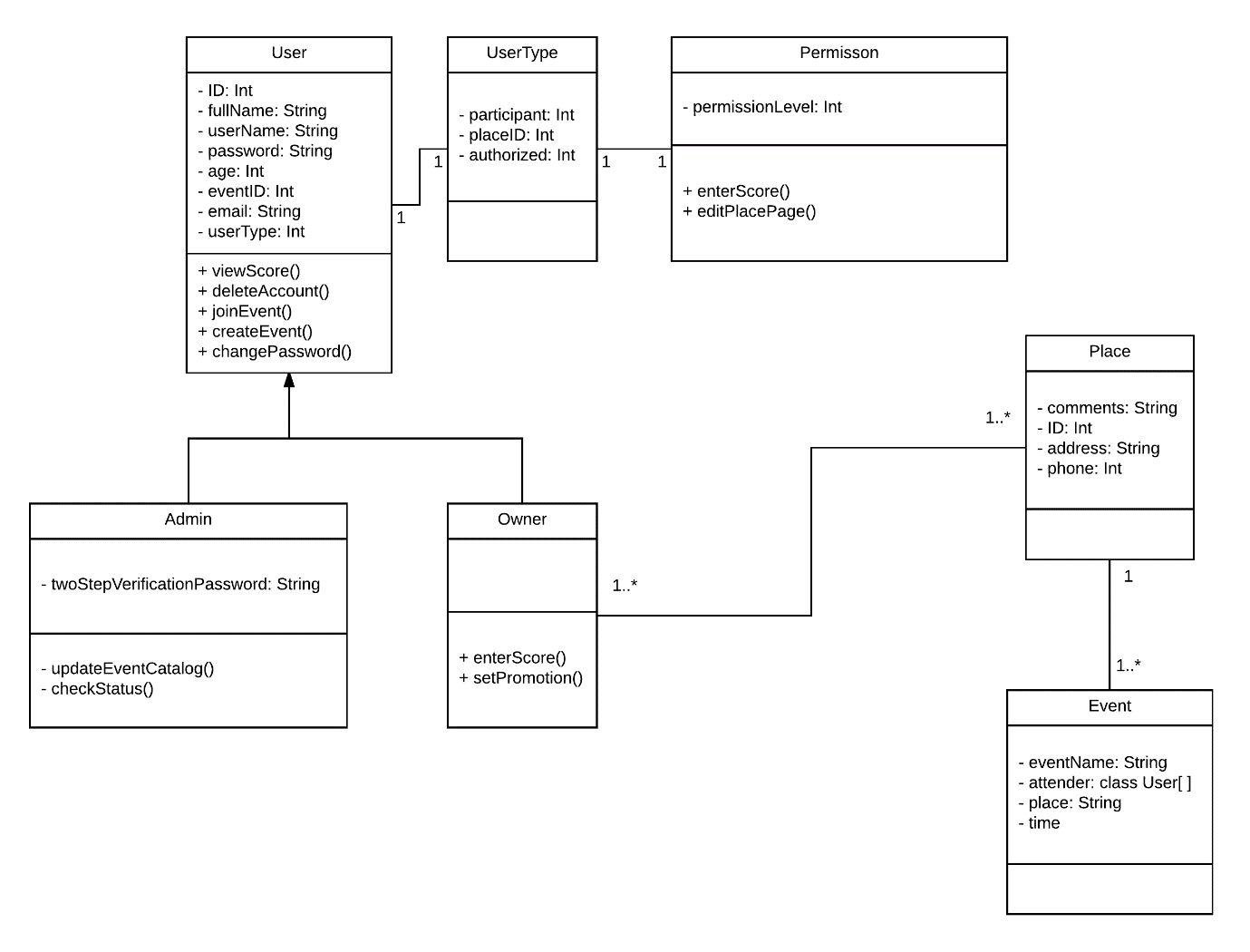
4. User adjust date and location, then waits for an opponent.

5. User views result of occurred events.

6. User comments, like or share the results.

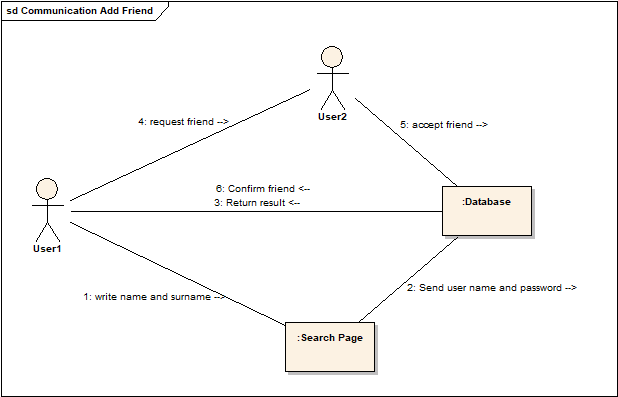
3.7. Class Diagrams

3.7.1. Class Diagram 1: Main Class Diagram

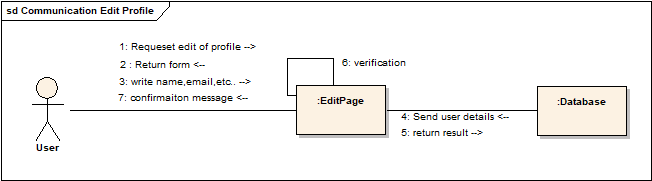


3.8. Communication Diagrams

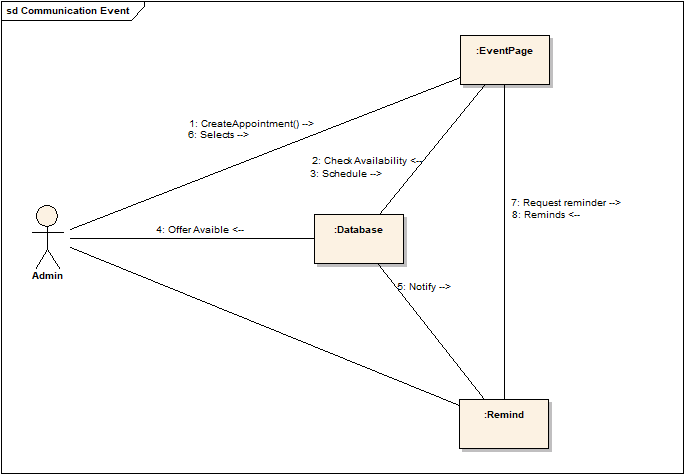
3.8.1. Communication Diagram 1: Add Friend

****

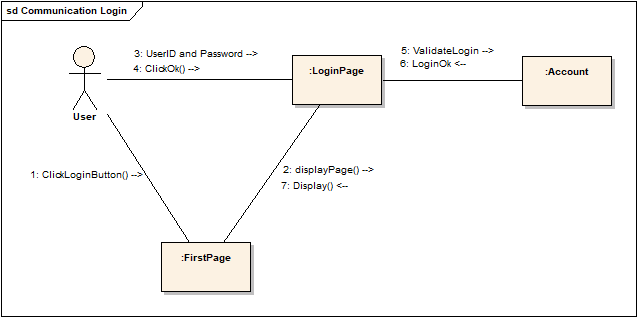
3.8.2. Communication Diagram 2: Edit Profile



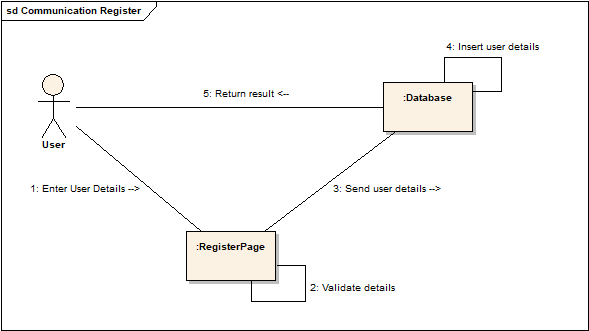
3.8.3. Communication Diagram 3: Event



3.8.4. Communication Diagram 4: Login

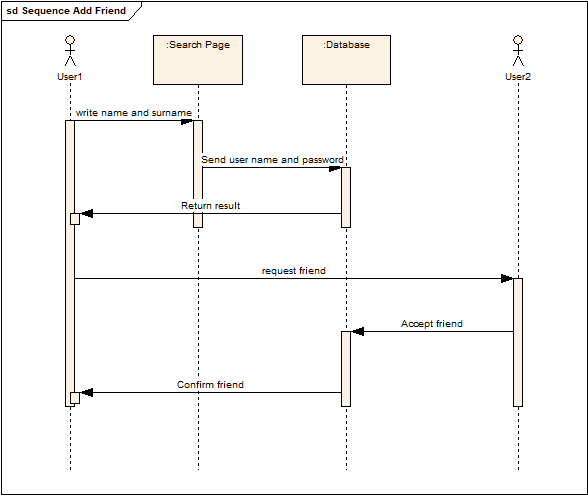


3.8.5. Communication Diagram 5: Register

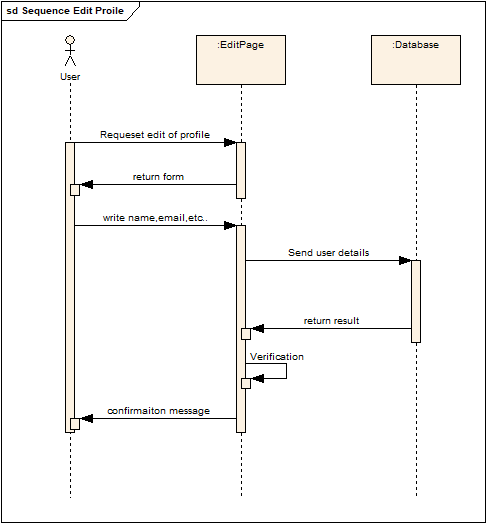


3.9. Sequence Diagrams

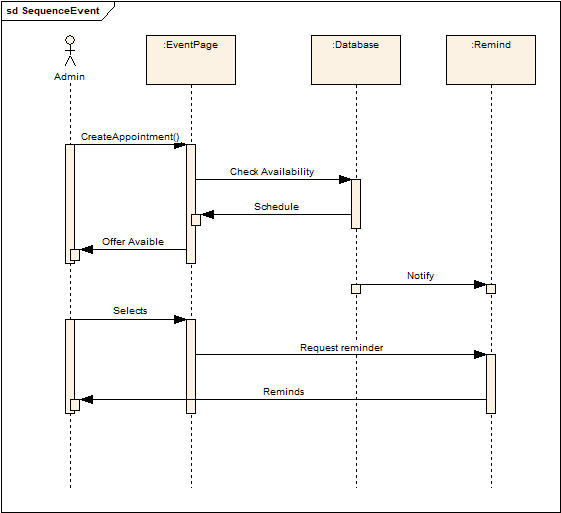
3.9.1. Sequence Diagram 1: Add Friend



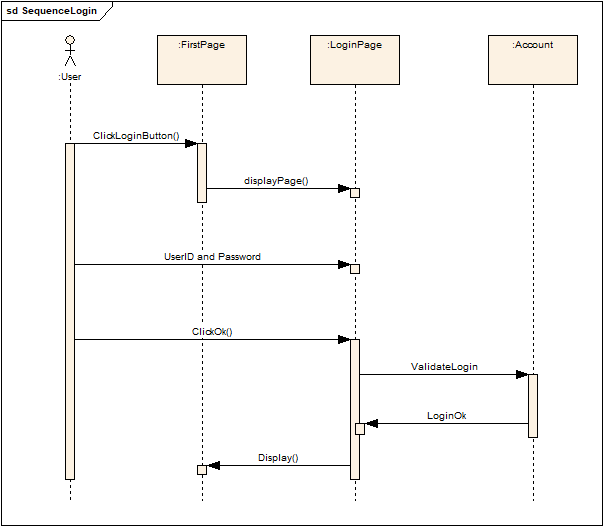
3.9.2. Sequence Diagram 2: Edit Profile



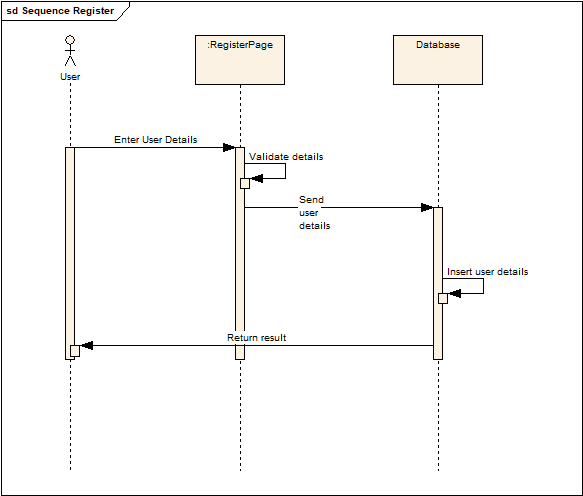
3.9.3. Sequence Diagram 3: Event



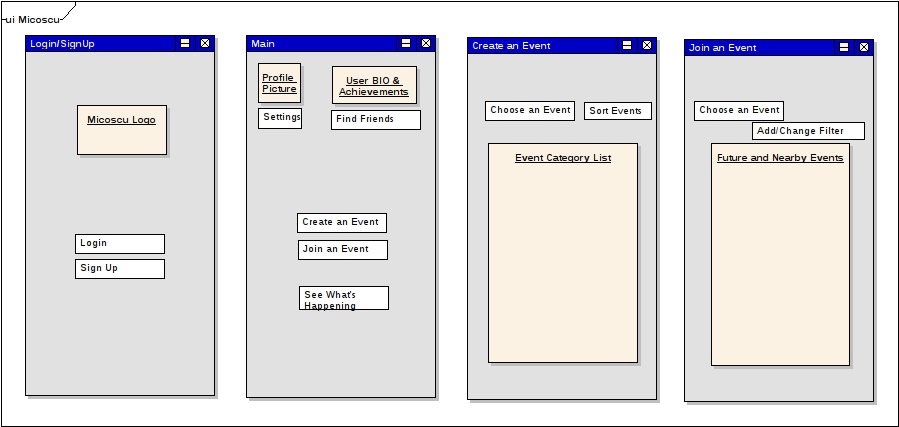
3.9.4. Sequence Diagram 4: Login



3.9.5. Sequence Diagram 5: Register



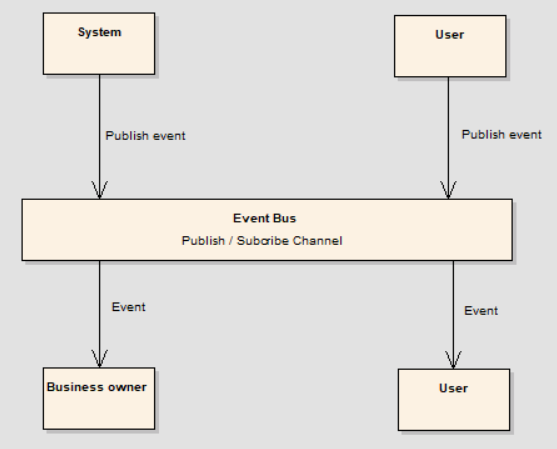
3.10. User Interfaces



4. Architectural Pattern

We decided to do with ***Eventbus Pattern***. Because the following causes consist of the items we need in our project

* **Simple yet powerful:** EventBus is a tiny library with an API that is super easy to learn. Nevertheless, your software architecture may great benefit by decoupling components: Subscribers do not have know about senders, when using events.
* **Battle tested:** EventBus is one of the most used Android libraries: thousands of apps use EventBus including very popular ones. Over a billion app installs speak for themselves.
* **High Performance:** Especially on Android, performance matters. EventBus was profiled and optimized a lot; probably making it the fastest solution of its kind.
* **Convenient Annotation based API** (without sacrificing performance)**:** Simply put the @Subscribe annotation to your subscriber methods. Because of a build time indexing of annotations, EventBus does not need to do annotation reflection during your app’s run time, which is very slow on Android.
* **Android main thread delivery:** When interacting with the UI, EventBus can deliver events in the main thread regardless how an event was posted.
* **Background thread delivery:** If your subscriber does long running tasks, EventBus can also use background threads to avoid UI blocking.
* **Event & Subscriber inheritance:** In EventBus, the object oriented paradigm apply to event and subscriber classes. Let’s say event class A is the superclass of B. Posted events of type B will also be posted to subscribers interested in A. Similarly the inheritance of subscriber classes are considered.
* **Zero configuration:** You can get started immediately using a default EventBus instance available from anywhere in your code.
* **Configurable:** To tweak EventBus to your requirements, you can adjust its behavior using the builder pattern.



5. Conclusion

This report introduced a simple social media platform for making people sense of entertainment, more real, more social meanwhile offers business owners to expand their business. Preparing gaming environments all over the world is more expensive than simple online board game app, but it is important to be active and stay outdoor