Player 2

Project documentation

Student(s):

* Gavrilă Tiberiu-Valentin

**Group: 30431**

**Contents**

[I Project specification](#_heading=h.gjdgxs) **3**

[1.1 Domain Model Diagram](#_heading=h.30j0zll) 3

[II Use-Case model](#_heading=h.1fob9te) **3**

[2.1 Users and stakeholders](#_heading=h.f2irg1azp7rm) 4

[2.2 Use-Case identification](#_heading=h.3znysh7) 4

[2.3 UML Use-Case diagram](#_heading=h.2et92p0) 4

[III Architectural design](#_heading=h.tyjcwt) **5**

[3.1 Conceptual architecture](#_heading=h.3dy6vkm) 5

[3.2 Package diagram](#_heading=h.1t3h5sf) 5

[3.3 Class diagram](#_heading=h.4d34og8) 5

[3.4 Database (E-R/Data model) diagram](#_heading=h.2s8eyo1) 6

[3.5 Sequence diagram](#_heading=h.17dp8vu) 6

[3.6 Activity diagram](#_heading=h.3rdcrjn) 6

[IV Supplementary specifications](#_heading=h.26in1rg) **6**

[4.1 Non-functional requirements](#_heading=h.lnxbz9) 6

[4.2 Design constraints](#_heading=h.35nkun2) 6

[V Testing](#_heading=h.1ksv4uv) **7**

[5.1 Testing methods/frameworks](#_heading=h.44sinio) 7

[5.2 Future improvements](#_heading=h.2jxsxqh) 7

[VI Bibliography](#_heading=h.z337ya) **7**

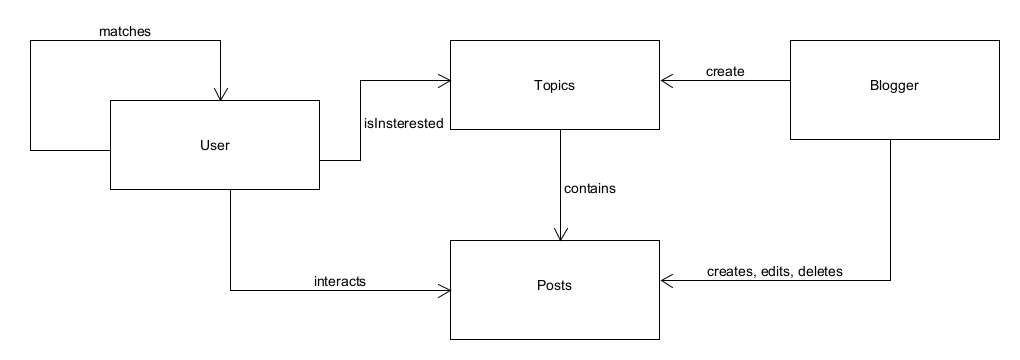
# I Project specification

Player 2 is a dating application targeted towards geeks, otakus, nerds, and gamers. Since there are many cliques and fandoms out there, basic dating app functionality (describing yourself, stating your preferences, and finding possible partners) is not sufficient to effectively bring people together, so, the user can be “interested” in a “topic” (for example: a video game franchise or a new movie). A topic is comparable to a blog page which is created and maintained by a third-party. Main users can interact with blog posts, and interactions are partially visible to other users.

Users must be able to create a profile and upload pictures, show owned games/book/manga, create an avatar (for the situations where they are anonymous), and customize the appearance of their profile, not only the contents. Search for potential matches, have matches and message them.

“Bloggers” (or “cliques”) must be able to create and delete posts, and check for interaction statistics per post or in a period of time (for the whole topic).

## 1.1 Domain Model Diagram

**

# II Use-Case model

Being a dating app, some use cases should be common for this type of application.

## 2.1 Users and stakeholders

User are represented by adults (18-35) with a keen interest in video games, novels, comic books, manga, anime, or other pieces of media or entertainment considered

Stakeholders are all represented by companies that own an IP (intellectual property) of interest for our target demographic (users). + competition, ourselves.

*<Intentionally left blank>*

## 2.2 Use-Case identification

**Use case name**: message & view already matched profiles

**Level:** subfunction

**Main actor**: user

**Main success scenario**: The user finds them in the matches list and can view their profile, and send them messages

**Extension**: if the matched profile has been deleted the conversation history is available, but the user can dens new messages

**Use case name**: search a blog post and see post interactions

**Level:** subfunction

**Main actor**: user

**Main success scenario**: the user has first to find and search the topic, and then search in the topic’s post., from which the post interaction should be *partially* visible.

**Extension**: if the post doesn’t have any interaction such the user will be prompted to interact.

**Use case name**: interaction statistics for a post

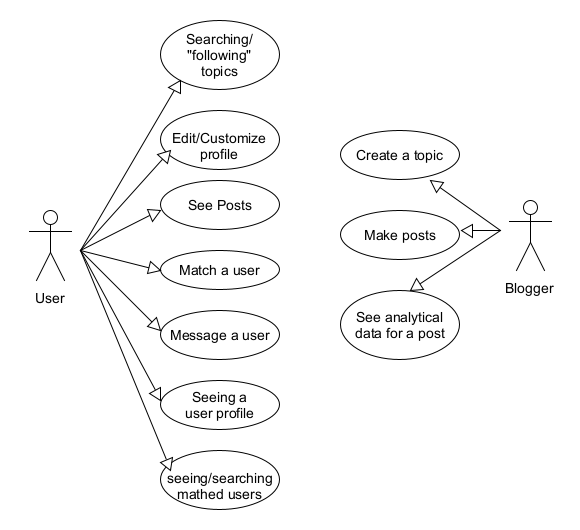
**Level:** subfunction

**Main actor**: “blogger”

**Main success scenario**: the “blogger” selects one of its posts to see analytical and statistical data

**Extension**: the “blogger” may want to see the data as different types of graphs

## 2.3 UML Use-Case diagram



# III Architectural design

“Player 2” needs to support basic instant messaging features and be reliable.

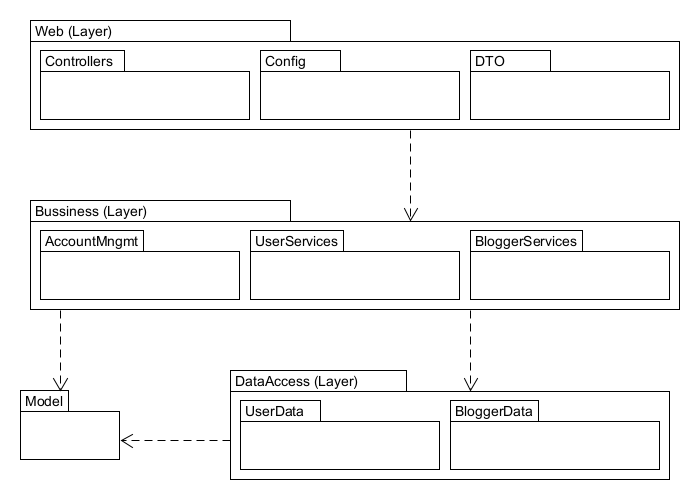
## 3.1 Conceptual architecture

Type of application: Web.

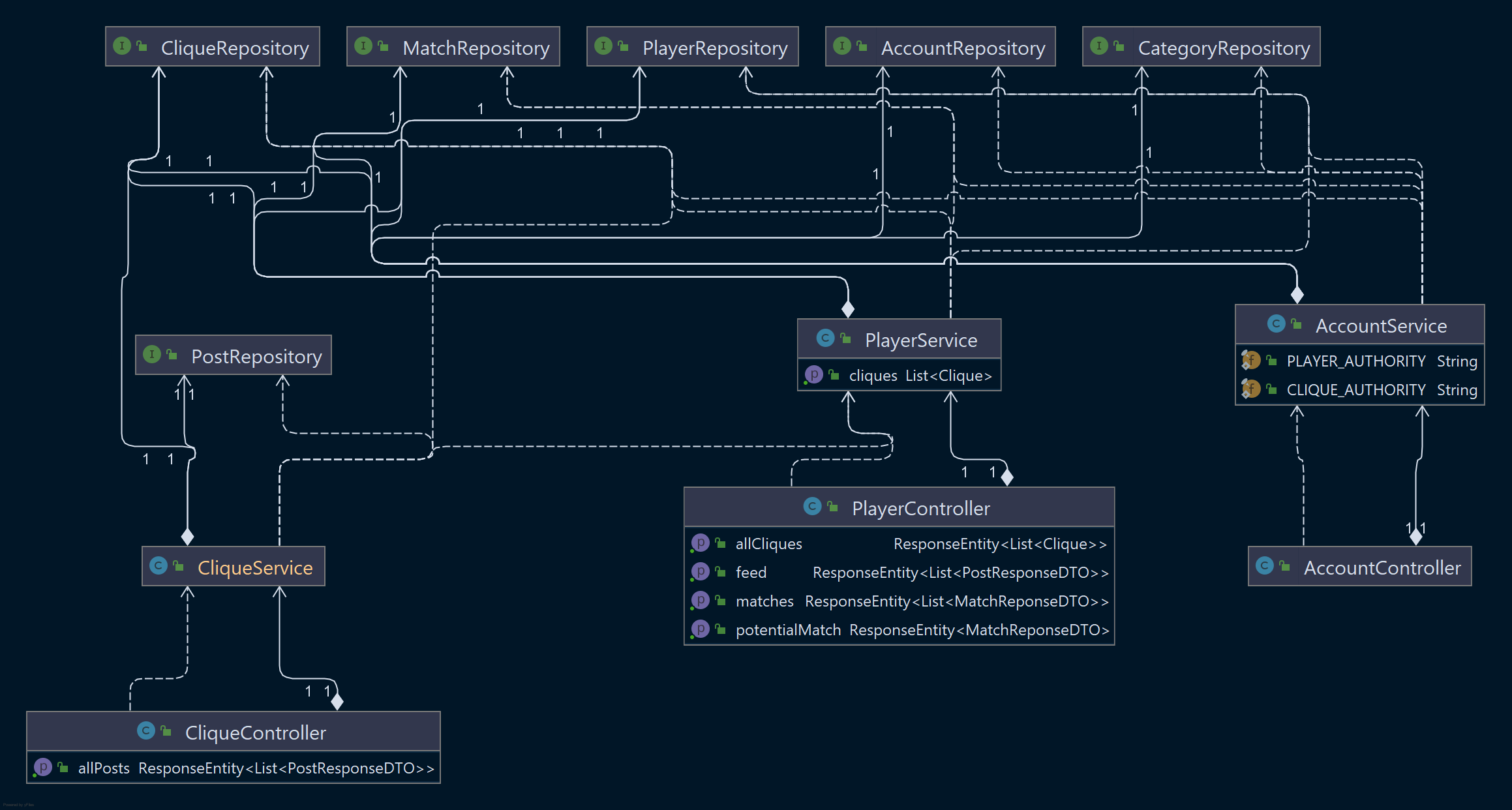
Architecture: layered, client-server, with pub/sub for instant messaging.

Since this is a web application, a client-server is the best fit for the overall architecture. We need to store a database of users and connect them with each other. And, within this overall architecture various design patters should be used for specific features.

## 3.2 Package diagram

**

## 3.3 Class diagram



## 3.4 Database (E-R/Data model) diagram

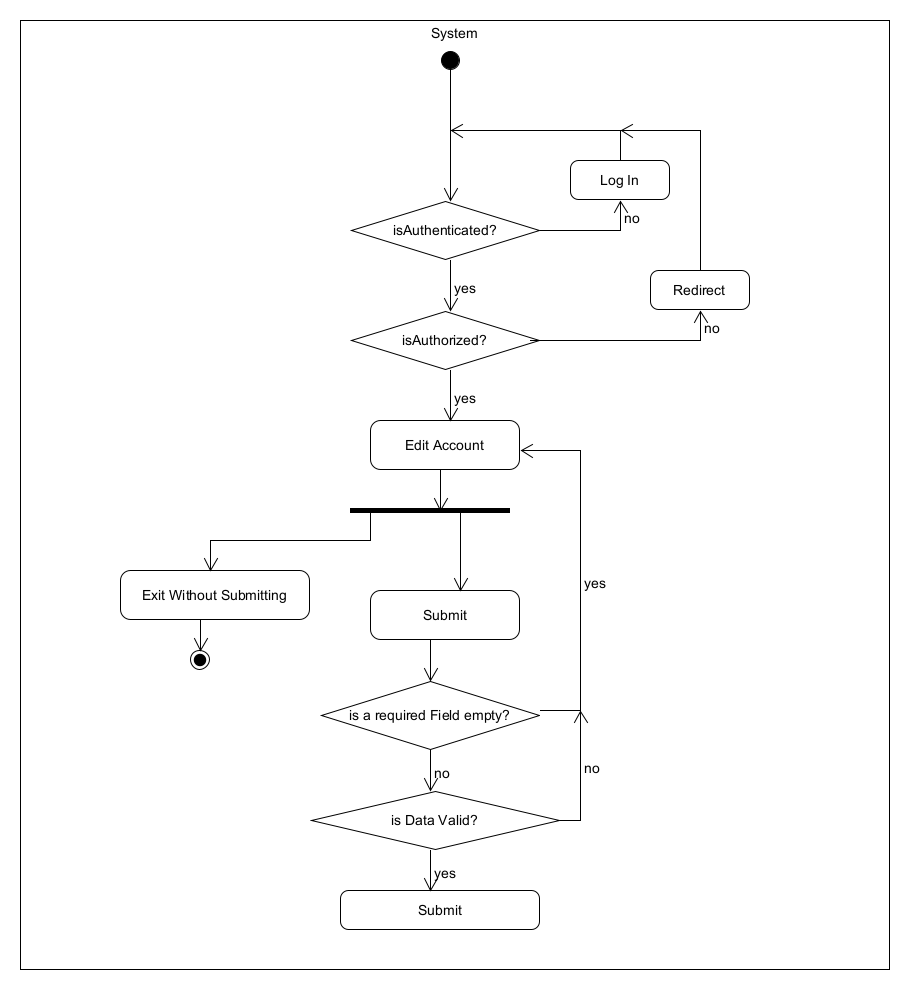
Diagram

Description automatically generated

## 3.5 Sequence diagram

## 

## 3.6 Activity diagram



# IV Supplementary specifications

*< Se va scrie o mica introducere./>*

## 4.1 Non-functional requirements

Inclusiveness and anonymity: most users (within the niches we target) should feel welcomed and comfortable in the online environment and community.

Scalability: should be able to accommodate a large user-base.

Availability & accessibility across devices: must be able to run on most web devices reliably, making it responsive for phones is a strong consideration.

## 4.2 Design constraints

Language and Full Stack Framework: C# with .NET.

Front-End Framework: Angular.

Database: Microsoft SQL Server (with Azure Cosmos DB for data that isn’t fit for the rigid structure of SQL)

# V Testing

*< Se va discuta la laborator./>*

## 5.1 Testing methods/frameworks

## 5.2 Future improvements

# VI Bibliography