**Assignment 1**

**Szymon Pawlica**

**R00187226**

**Services**

The design of a microservice-based system that would meet the requirements would need to consist of the following microservices and ensure smooth communication between them.

A **Registration Authentication Service**, this microservice will allow users to register an account within the system, data will be validated and encrypted. Information stored will include the user’s login, email, password, and IP address.

A **Login Authentication Service**, this microservice will validate user login information and allow users to log into the system with their registered account. This will store information such as the username, password, and session data.

A **Game Catalogue Service**, this microservice will allow for the listing of all games available in the store. For each game, information such as the title, price, release date, description, etc., will be stored in a database.

A **Game Download Service**, this microservice will hold information about the download status of games. The information stored will along with the download status include the download URL.

An **Achievement Service**, this microservice will hold the information on achievement progress, it could also display the % of players that earned each achievement to see which are the most difficult to acquire. The information would include the achievement name, progress, completion date etc.

A **Point Service**, this microservice will allow for the earning and spending of points which can later be used to purchase vouchers or offer profile/in-game customization. Information stored would include point cost, effect of purchase, availability etc.

A **Profile Service**, this microservice will allow users to personalise their experience through preference settings and for the possibility of adding friends. The information stored would include the users name, preferences, games owned, points, status etc.

A **Discovery Service**, this microservice will allow for users to get a personalised experience when browsing for games as it will present games that match the genre of games owned/played by the user. This service will store information like the game genre, playtime, last played etc.

A **News Service**, this microservice will provide the latest of interest gaming news, the news will be filtered by genre and news type e.g., upcoming releases and updates. Information stored will include the news title, content and date posted.

A **User Content Service**, this microservice will allow users to post their own content to games that support modifications to enhance their experience. Stored information will contain the title, description, creator/s, download URL etc.

**Approach**

With this general overview of the required microservices, this is how they are going communicate with each other and how the system is going to function.

For a user to create a profile they will have to complete a registration form in their browser, the details provided by the user will be encrypted and checked to make sure the username does not match any other existing users via the **Registration Authentication Service**. From there whenever the user wants to log in to use the system their information will be run through the **Login Authentication Service** which will ensure the users details are correct before letting them proceed and interact with the system as their user profile, both services will communicate through gRPC. From there they can edit their profile preferences via the **Profile Service**, they will also be able to create their own game modification content for games that support it via the **User Content Service** which will publish this content for the rest of the users interested in using it.

A user will be able to interact with the **Game Catalogue Service** and purchase any games they are interested in, each game will hold relevant information such as the games release date, publisher, developer, information and reviews. Once a game is purchased users should be redirected to a **Payment Authentication Service** where they will fill in their payment details which will be encrypted for security and have them verified before their purchase is accepted, taking any discounts into account. If the purchase is successful, the user should have their game library and points owned updated through their **Profile Service**, the game should also be available for download via the **Game Download Service** immediately after purchase, this communication will be done through gRPC to ensure the user can access their game immediately. Once a purchase if complete additional messages will be sent to other services through the use of RabbitMQ such as the **Achievement Service** which will update the games achievement completion rates of the game as there is a new player with zero achievements completed; the **Discovery** and **News Services** to update them based on the users game genre interests and to see what other players of that game are interested in.

The **Game Download Service** will only be made available for a user if they have purchased the game this will be ensured through a quick authentication which will check if the game the user is attempting to download is listed in their “owned games”. This will to some extent prevent users from providing others with a download link to the game.

The **Point Service** will allow users to purchase unique profile modifications such as animated profile pictures, special emotes which can be used in messages with friends or in groups, vouchers which would provide discounts for specific games. This service would require communication with the **Profile Service** to ensure that they have the minimum required number of points to complete their purchase. The communication between these two services will be done via gRPC.

The **Profile Service** will allow users to manage their profiles, this will include modifying what is visible by other players or friends, adding a custom profile comment, recent game activity, an achievement showcase, friends list, user content published, and profile comments made by others. It will also allow for preference modifications to allow the user to opt out of game betas, change their theme, change their online status.

The **Discovery Service** will suggest games to the user based on their recent purchases, owned games, recent games played. From their games with similar genres or from the same developers will be presented to the user, each game can be listed as ‘not interested’ if this is done the game won’t appear on the user’s discovery for a month.

The **News Service** will allow users to read about current and upcoming game events/releases, it will feature a comments section along with a like/dislike feature for each post. It will also gather information from the users that interact with it, this will include what types of content users are clicking on based on the genre of games they are interested in. This will allow for featuring post which are most popular/most liked today or over the last week or month, major releases will also be featured at the top of the page regardless of use preferences.

The **User Content Service** will allow users to publish their own creations/modification for games, users will need to upload their content which will then be checked for security issues before posting it into the public library of content. Each piece of content will show its information, creators, release date, reviews and a comments section. This will also feature a most subscribed and most liked feature to filter what user content is the most popular, additional filters will include a by game, release date as well as a search bar to search for a specific piece of user created content.

Here is a diagram representing the suggested architecture.

**Conclusion**

The system will consist largely of single service – single database services to ensure loose coupling of services as well as tight cohesion [1]. The registration and login services access different databases to follow the single responsibility principle, registration being data storage focused and login focused on authentication [2]. Communication between services that require communication with no more than one other service will use gRPC these include the registration authentication procedure as well as the game/points purchasing services to ensure short wait times, whereas services that require communication with multiple other services will use RabbitMQ in a producer consumer like fashion [3].

# References

|  |  |
| --- | --- |
| [1] | "Difference Between Cohesion and Coupling," [Online]. Available: https://www.baeldung.com/cs/cohesion-vs-coupling. |
| [2] | "Principles for Microservice Design: Think IDEALS, Rather than SOLID," [Online]. Available: https://www.infoq.com/articles/microservices-design-ideals/. |
| [3] | "Pub-Sub vs. Message Queues," [Online]. Available: https://www.baeldung.com/pub-sub-vs-message-queues. |