2 System Overview

This section should describe the overall structure of your software system. Think of it as the strategy for how you will build the system. An architectural "layer" is the top-level logical view, or an abstraction, of your design. Layers should be composed of related elements of similar capabilities, and should be highly independent of other layers, but should have very clearly defined interfaces and interactions with other layers. Each layer should be identified individually and should be unique as to its function and purpose within the system. This section should also contain the high-level block diagram of the layers, as shown in the example below, as well as detailed descriptions of the functions of each layer.

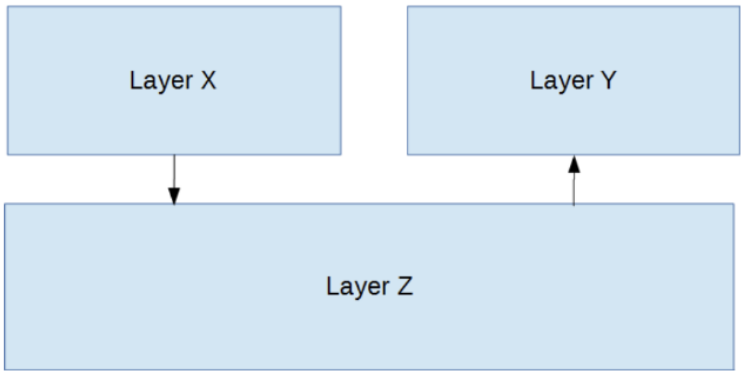


Figure 1: A simple architectural layer diagram

2.1 Layer X Description

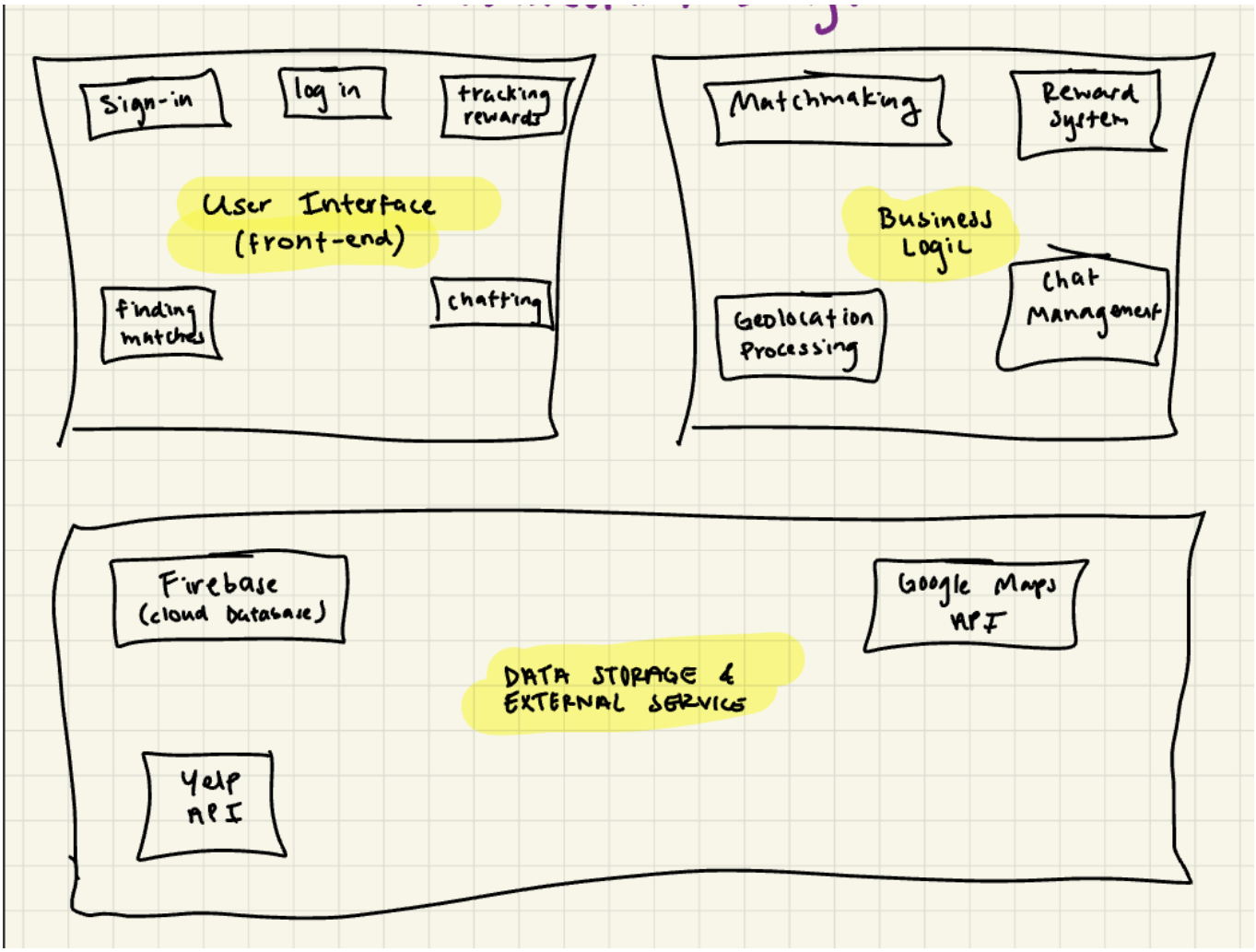
Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.

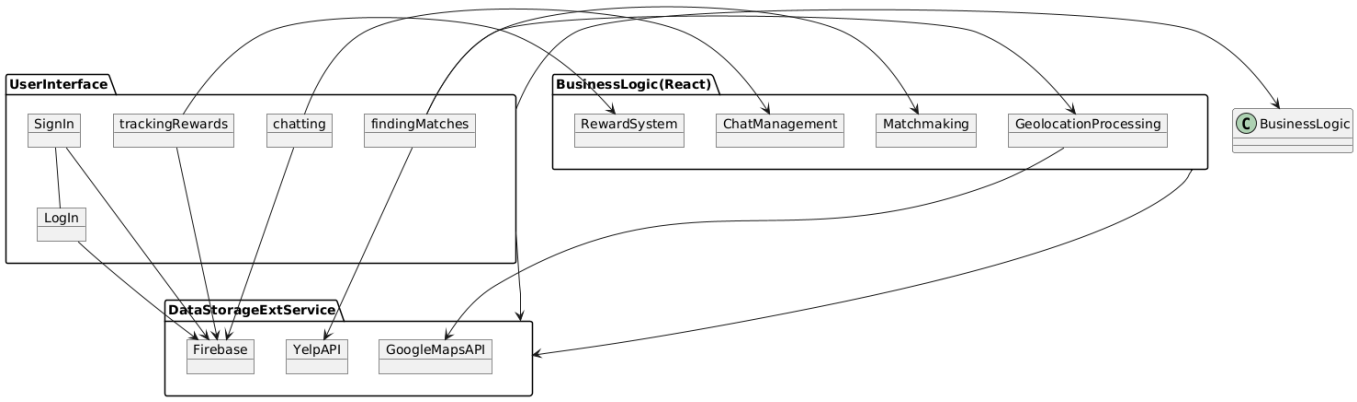
2.2 Layer Y Description

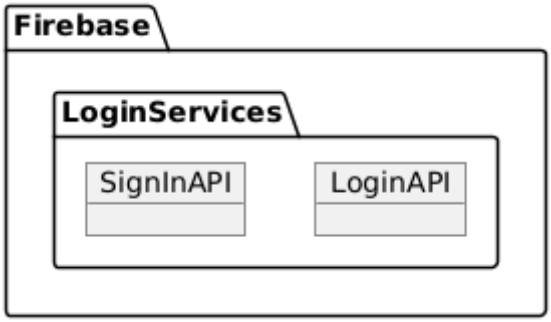
Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.

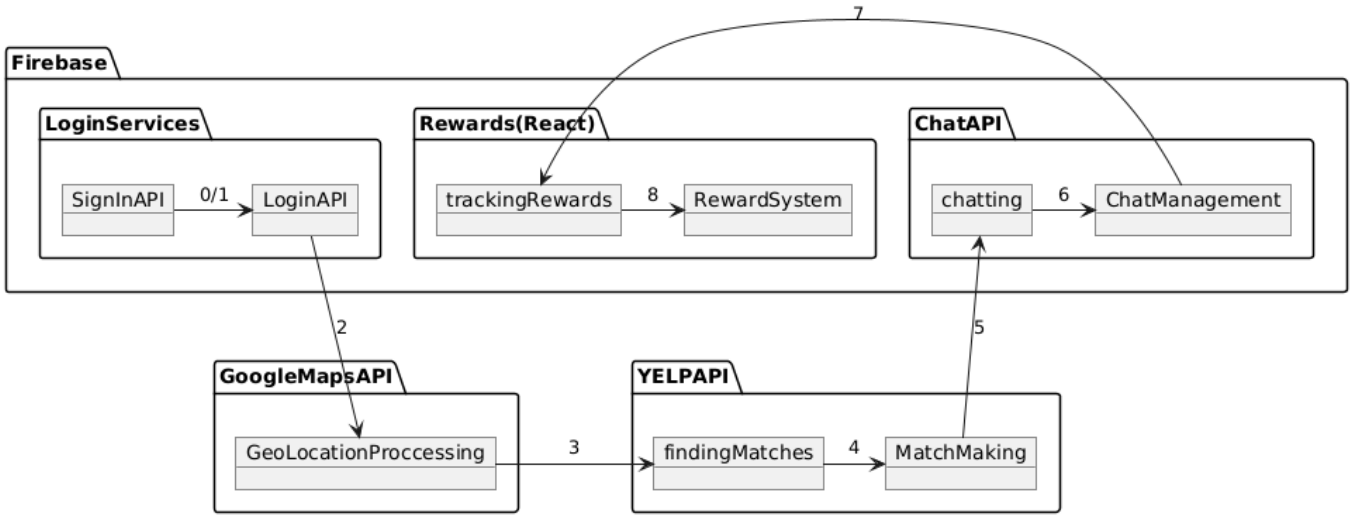
2.3 Layer Z Description

Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.


Sign-in
log in
tracking rewards
User Interface (front-end)
finding matches
chatting
Matchmaking
Reward System
Business Logic
Geolocation Processing
Chat Management
Firebase (cloud database)
Google Maps API
DATA STORAGE & EXTERNAL SERVICES
Yelp API


UserInterface
SignIn
trackingRewards
chatting
findingMatches
LogIn
BusinessLogic(React)
RewardSystem
ChatManagement
Matchmaking
GeolocationProcessing
BusinessLogic
DataStorageExtService
Firebase
YelpAPI
GoogleMapsAPI


Firebase
LoginServices
SignInAPI
LoginAPI



2.11 Firebase Layer

The Firebase Layer is where all objects that need storage will go. The separate systems for this are Login Services, Rewards system, and the ChatAPI, and the lower layers are defined by the specific services of the Application.

There will be a sign in(Sign up?) function, where Users can sign into an account with the application. Once these have been verified by the security measures, the operation will move on to data processing.

There will be a log in function, where Users can use already created passwords to log into their accounts.

The reward system will be the specific rewards and the algorithm behind the rewards. RewardSystem will be where the system to decide rewards will be, while the trackingRewards will be where the specific actions that influence the rewards will go.

The Chat Management system will be the system that uses AI or pre given API's (Unsure) to manage the Chat system. The chat system will be an API that allows for forum or DM style chatting between users of the app. chatting will only take into account people that have also seen the restaurants a user has already been matched to, and all chats will be managed by a ChatManagement tool.

2.21 GoogleMapsAPI Layer

This is the layer that is involved in the real time location services of the application. It will be accessed once a user has passed the login and sign in stage, and the information about the restaraunts found will be sent out to the YELPAPI.

Google Maps API will be used for the location tracking information.

2.31 YELPAPI Layer

The Yelp API layer is about the services regarding restaurants and the user's preferences.

The findingMatches function will be where the info about restaraunts that are close by the user in real time will be sent. This is then sent on to the MatchMaking software.

The Matchmaking function will be the algorithm that finds restaurants according to User preferences. It takes in the restaraunts found bu the findingMatches function, and then sends data to the ChatAPI layer with the info of the user's preferences to match them to other people with similar preferences.

OLD VERSION

2.1 User interface Layer

The user interface layer is the front end version of the application. This involves all parts that the user will interact with directly.

There will be a sign in(Sign up?) function, where Users can sign into an account with the application.

There will be a log in function, where Users can use already created passwords to log into their accounts.

There will be a finding matches function, which will find restaurants according to an algorithm based on the User's preferences.

There will be a tracking rewards function, which will be based on how much the application is used.
There will be a chatting function, where User's can talk to each other in either forum or direct messaging formats.

Sign in and log in functions will be closely related. Log in can only be accessed after sign in.

2.2 Business Logic Layer
The Matchmaking function will be the algorithm that finds restaurants according to User preferences.
The reward system will be the specific rewards and the algorithm behind the rewards.
The Geolocation processing system will be the system that converts the location information to data to be used by other functions.
The Chat Management system will be the system that uses AI or pre given API's (Unsure) to manage the Chat system.

2.3 Database storage and external database Layer
FireBase will be used for the storage of records and information for the application.
Yelp AI will be used to get information about restaurants.
Google Maps API will be used for the location tracking information.

4 X Layer Subsystems

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

4.1 Subsystem 1

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

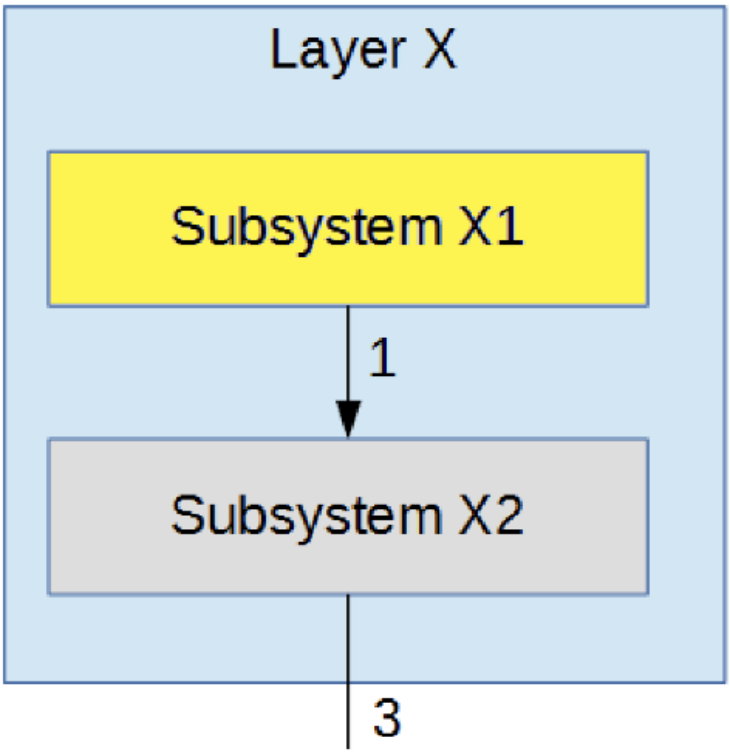


Figure 3: Example subsystem description diagram

4.1.1 Assumptions

Any assumptions made in the definition of the subsystem should be listed and described. Pay particular attention to assumptions concerning interfaces and interactions with other layers.

4.1.2 Responsibilities

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing

data elements will pass through this interface.

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

GeoLocationProcessing Subsystem

Assumptions

This service will only activate when the login and sign in checks can be verified. This service shall only work when the User has a working connection to the internet.

Responsibilities

This subsystem will take in the User's location with the GoogleMapsAPI. Then it will record the the restaurants that both are available with the YELPAPI and within a certain distance of the user, in an amount that will be predetermined by the user.

Subsystem Interfaces

ID	**Description**	**Inputs**	**Outputs**
#1	GeoLocationProcessing	True/False Login acceptance Location Range	Vector of Restaraunts(strings and IDs)