SwipeRight

Design

Luka Spaninks
Semester 6
RB03

Version 1

Preface

The complete design phase of the project SwipeRight can be found in this document. This file will evolve over time and can always be expanded upon.

Inhoud

1. Techstack

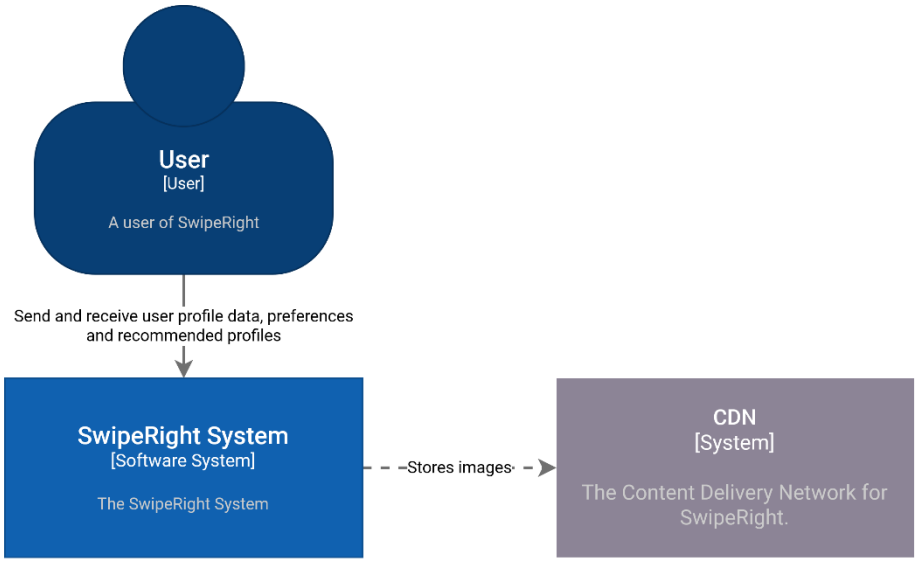
The technologies which will be used for the SwipeRight techstack are defined below.

Feature	Description	Motivation
Security	OAuth 2.0	Secure and widely used (industry standard)
Server script	Go	Performant, Scalability & Nice syntax
Server Framework	Gin	Gin is a performant and widely used backend framework for Go.
Client Technology	Flutter	Flutter is a performant cross-platform technology, It's used a lot for native development.
Client script	Dart	Dart is the language used in Flutter
Database 1	Apache Cassandra	Performant, Secure & highly scalable
Database 2	Redis	Flexible data structures, performant & simplicity
Server OS	Ubuntu	Backend programs should eventually be able to run in every environment
Client OS	Android	In the development phase my target OS is android because it is open, it might be extended to IOS in a later stage of production.

2. C4 Models

2.1 System Context

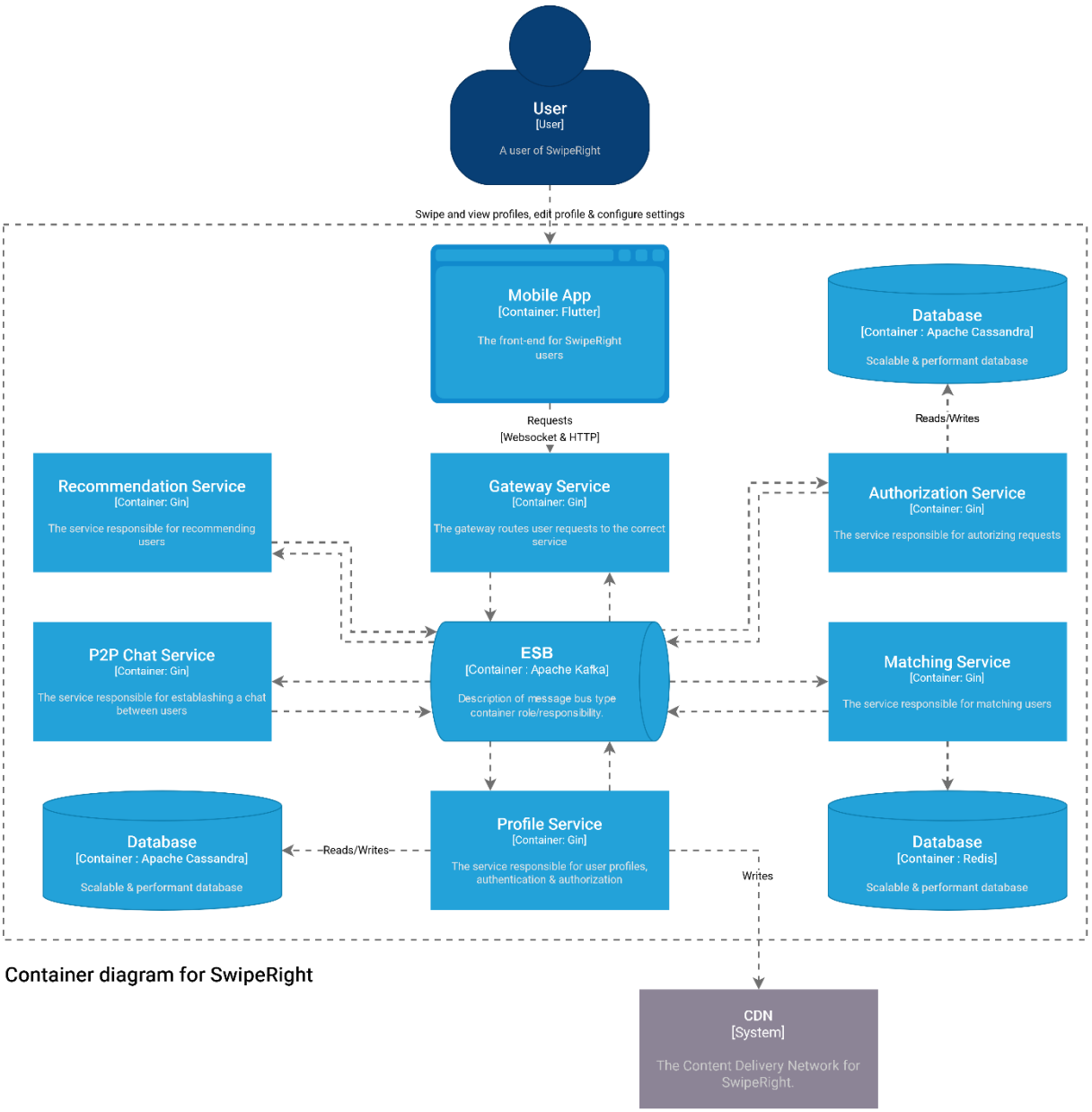
The idea is to work with a microservice architecture which is almost completely dependent on internal services. A system like a content delivery network should however be outsourced to an external system though. Vendors like amazon have servers everywhere in the world which helps a great deal with performance. This is a resource I unfortunately do not have access to.



System Context Diagram SwipeRight

2.2 Container diagram

The container diagram contains a gateway which connects the clients to the service bus. The service bus makes communication between all services possible.



Container diagram for SwipeRight

2.3 Component diagram

The component diagrams are per microservice.

3. Entity Relationship Diagrams

The entities are defined per microservice.

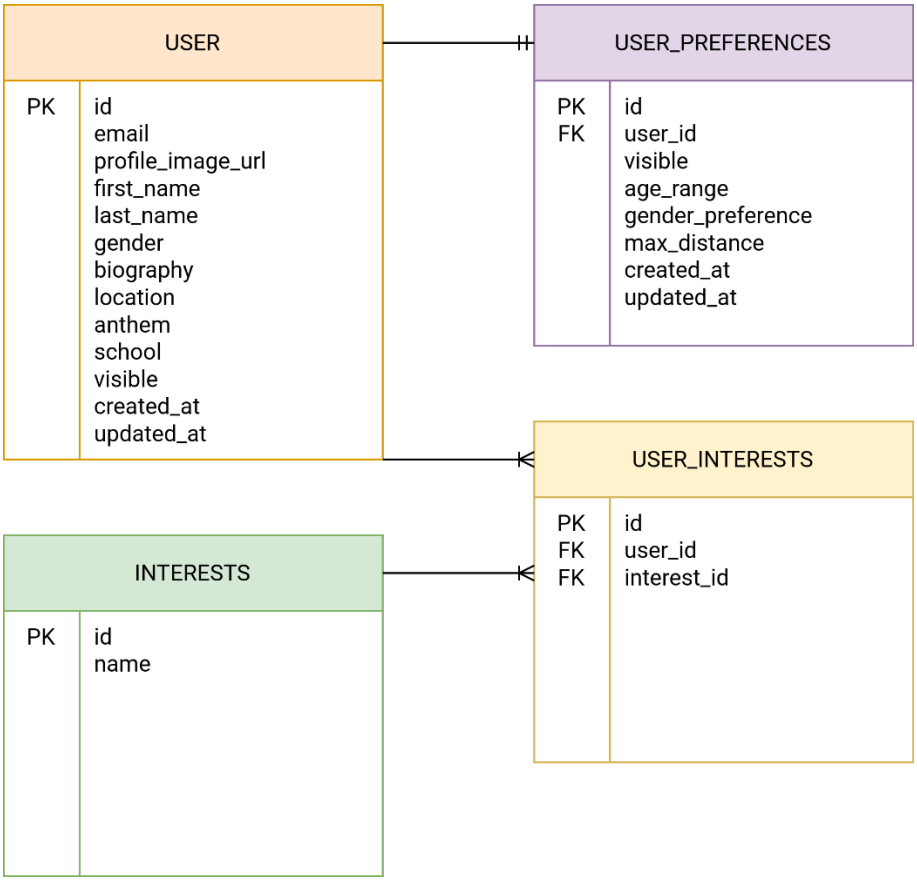
3.1 Authorization Service

The Authorization service will need to store user credentials. More will be added when the requirements are extended.

USER_CREDENTIALS	
PK FK	id user_id password created_at updated_at

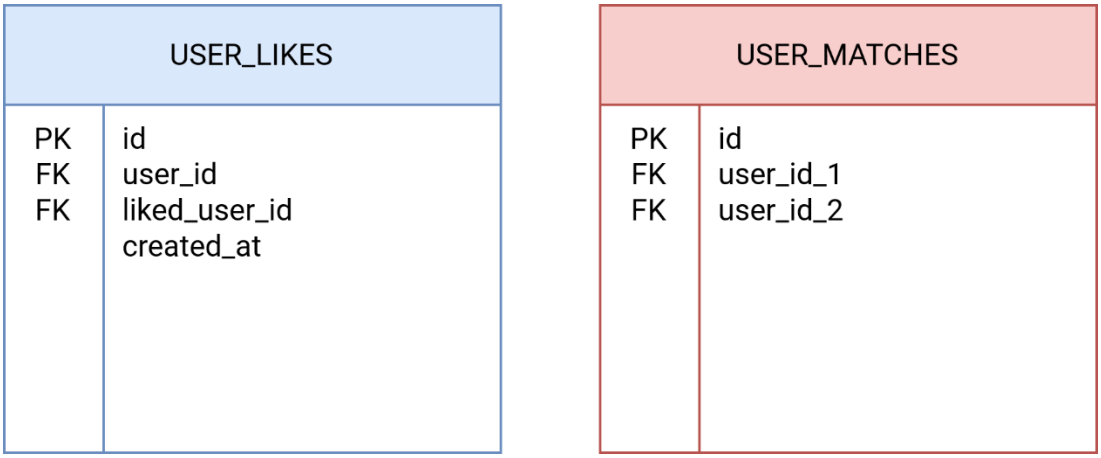
3.2 Profile Service

The profile service will store all user related data.



3.3 Matching Service

The matching service has the responsibility to keep track of the liked users and needs to update the USER_MATCHES when two users have liked each other.



4. User Interface

In this chapter the designs and standards are defined.

4.1 Design System

4.1.1 Typography

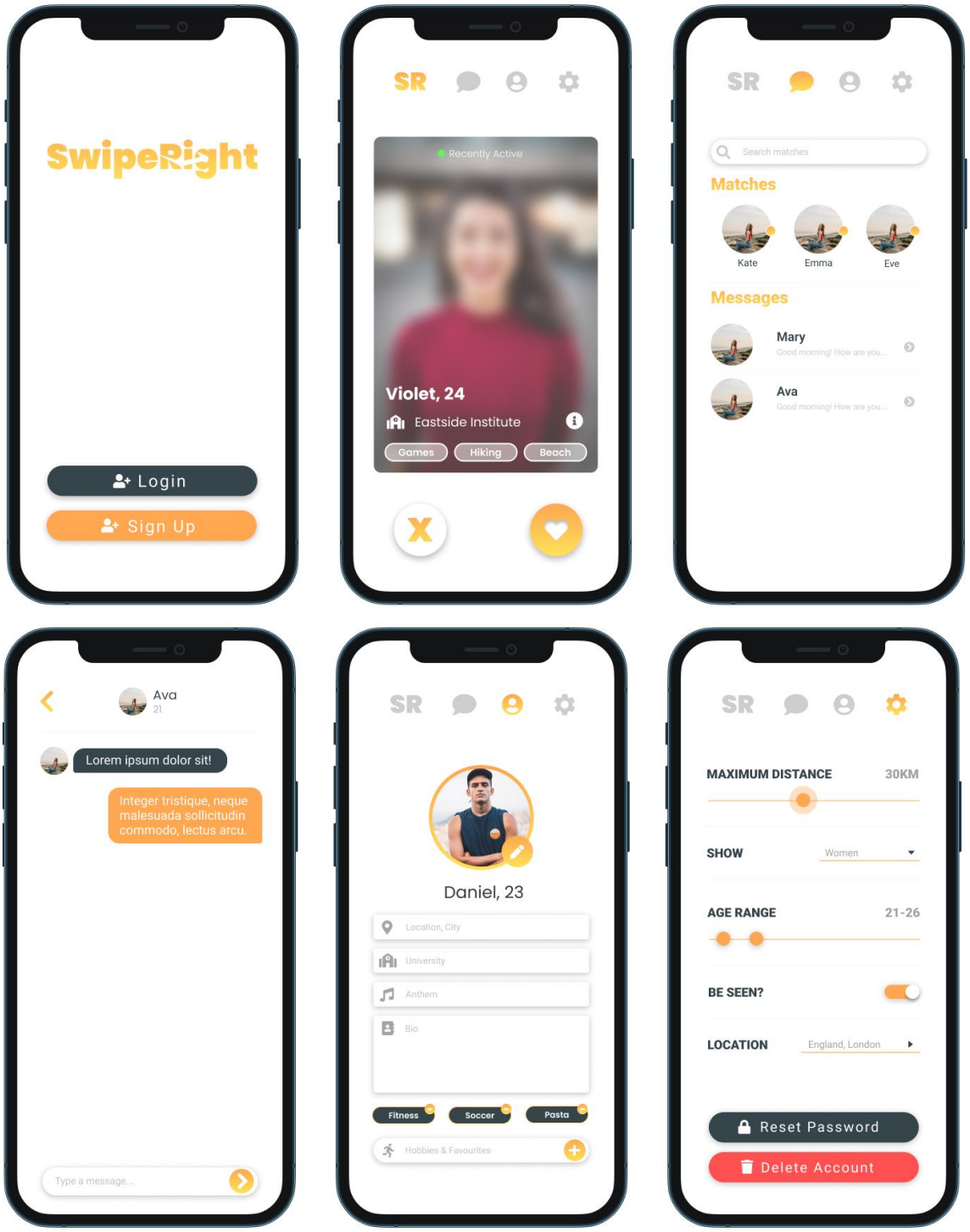
Name	Font
Logo Font	Poppins
App Font	Roboto

Name	Font Size (px)
XS	12
S	14
M	18
L	20
XL	24

4.1.2 Colors

Name	Color
Logo Gradient Top	#FFA751
Logo Gradient Bottom	#FFE259
App White	#FFFFFF
App Dark	#373737
App Grey	#CECECE
App Green	#54FF51
App Red	#FF5151

4.2 Mockups



*** Mockups are created with Figma*

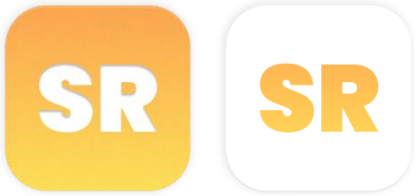
**** Pictures used are from Unsplash*

4.3 Assets

4.3.1 Logo

SwipeRight

4.3.2 App Icons



4.3.3 Like & Dislike Buttons

4.3.4 Default Buttons