Ride Sharing Website

COMPX374-20B (HAM) Software Engineering Industry Project

Deliverable II

Software Design Specification

12th September 2020

Team Members:

Aaron Win

Lysa Phan

Victor Yao

Ryan Good

THE UNIVERSITY OF

WAIKATO

Te Whare Wānanga o Waikato

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this report is to outline in detail the software architecture and the design for Ride Sharing Website. This report presents several views of the system's design, which depict the interaction between system components and illustrate the data-flow of the system. All the decisions on software architecture and design are captured and conveyed to show a clear understanding of the system.

1.2 Scope

This Software Design Specification defines how the software architecture and design will accomplish the functional and non-functional requirements that have been mentioned in Software Requirement Specification report.

1.3 Intended Audience

This report is written on a technical level and is used by the software development team as a definition of the design to implement the Ride Sharing Website. It will be sent to the customer for review and future development purpose. This report will also be evaluated by the convenors of the course Software Engineering Industry Project.

2. Changes in Requirement

There have been no substantial or noteworthy changes in both functional and non-functional requirements of the project since the submission of Deliverable I-Software Requirements Specification.

3. Software Architecture

3.1 Component Diagram
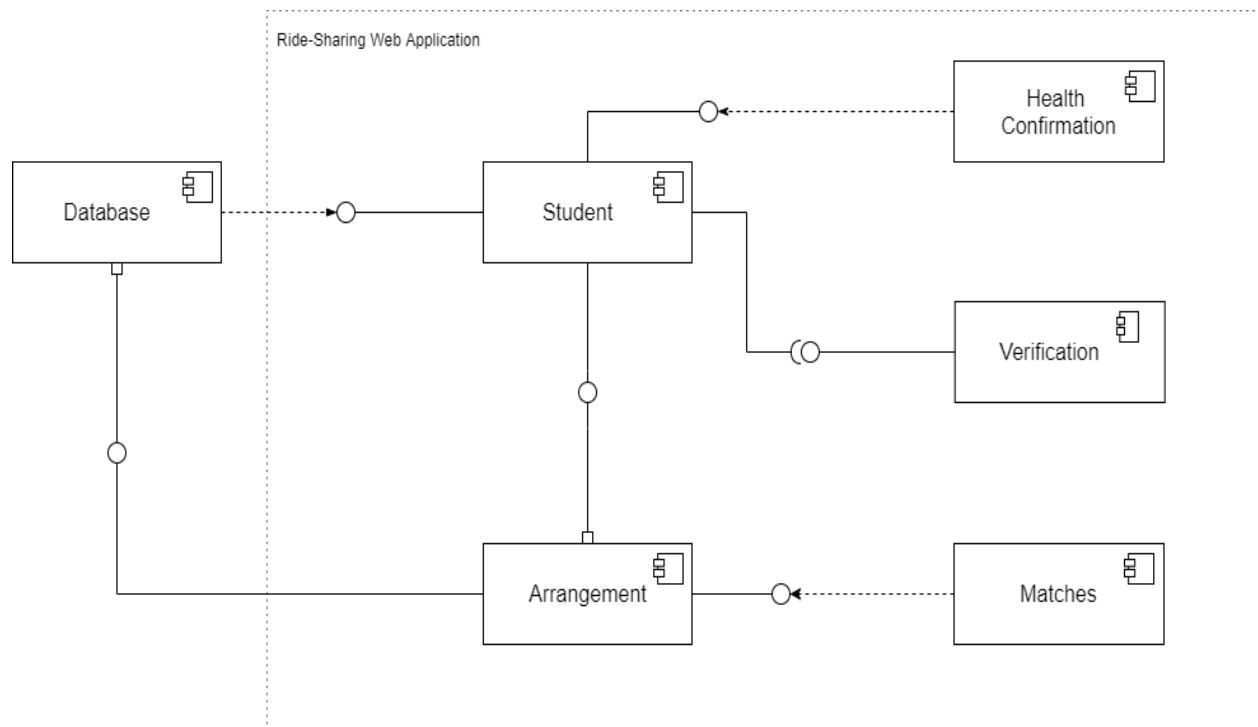
Database – The Database component stores the data pertaining to the users and the ride-sharing arrangement. It gives data to the student's profile which will be display when student offers or requests a ride. It also stores and retrieves data for the ride-sharing arrangement.

Student - The Student component stores and sends out the necessary data and values that relate to the user and their intended use of the web application. When the user creates an arrangement to share ride, the Student component will display the student's name and the ride's details in the arrangement.

Verification - The Verification component allows user to log in to the website securely. This is facilitated by Facebook, so the user is required to use their Facebook account to gain access to the web page. The user must verify their account first, before they can proceed further with the web application.

Health Confirmation - The Health Confirmation component is a requirement that needs to be filled out by the user every time they log in to the website. This ensures the user is healthy enough to share ride without infecting others.

Arrangement - The Arrangement component conducts and facilitates the ride-sharing. Each arrangement features one driver and one passenger. At this stage, the website only allows one passenger account to join a ride. The option of having more passengers in one ride can be arranged via users discussing between themselves. The Arrangement will store the location, user details and timeframe to conduct the ride-sharing.

Matches - The Matches component displays all the available rides that meet the criteria of user. It will include the suitable timeframe of the ride, the physical distance from the user to the driver, and the distance from there to the actual destination. When the user chooses to accept one of the matches, they can have further discussion about the ride with each other.

3.2 Deployment Diagram
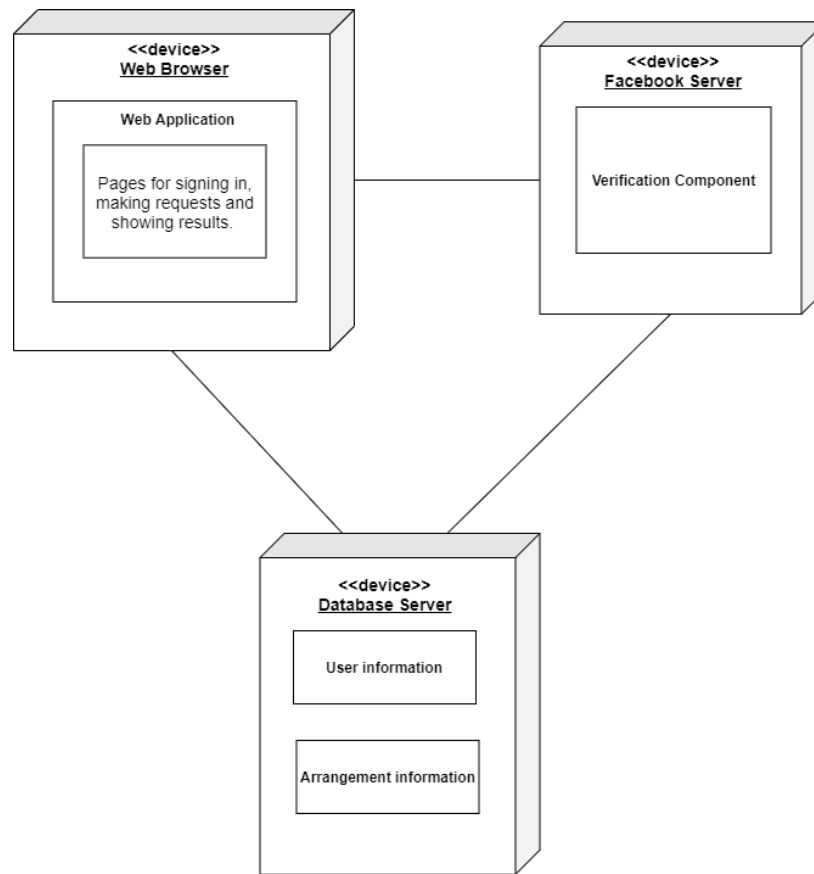
Web Browser - Stores the web application including the pages for signing in, initiating arrangements and displaying results. The web browser is associated with the device of the user (e.g. laptop or phone).

Facebook Server - External entity which is required to let user verify their Facebook account. The user can only proceed further on the web application if the verification is successful.
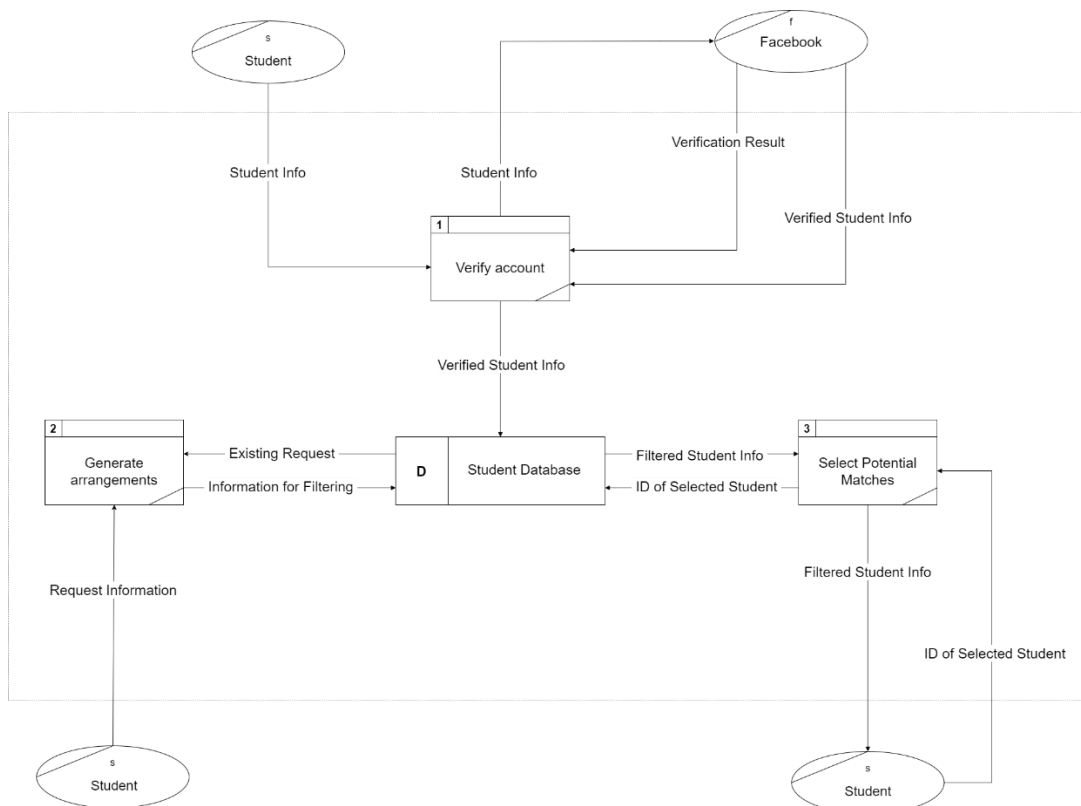
Database Server - Stores the details of all users and their arrangements. The information stored in the database is the user's name, the role, timeframe, origin location-destination, party number, contact number and number of people already in one ride.

4. Detailed Design

4.1 Data-Flow Diagram Level 0
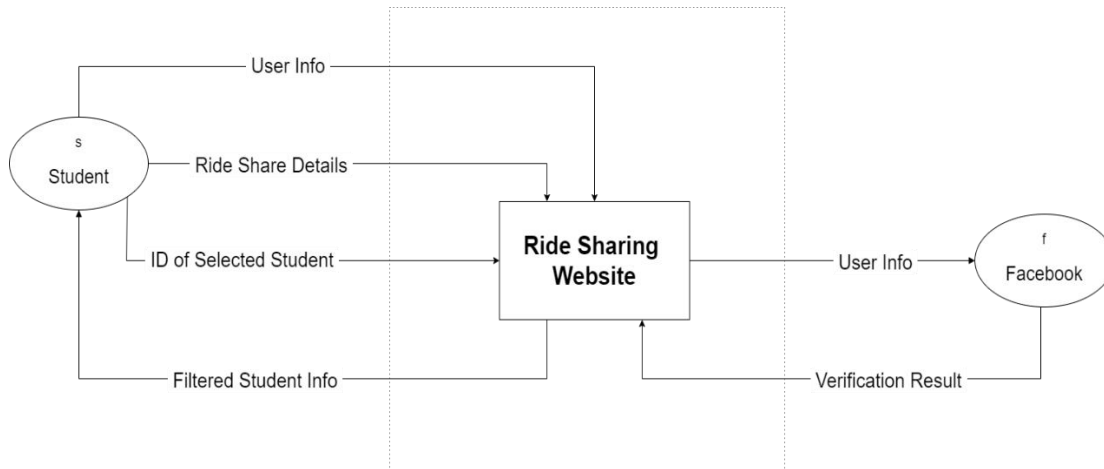
4.2 Data-Flow Diagram Level 1

Process 1 verifies user account, which allows the user to log into the web page by using a third party which is a trusted and secure service.

Data flows in:

- From external entity **Student s**: *Student Info* such as email and password.
- From external entity **Facebook f** server: *Verified Student Info* and *Verification Result.*

Data flows out:

- To **data store D**: *Verified User Info* from **Facebook f**.

Process 2 takes the request information to the database. It goes through a list of existing requests taken from the database. and compares the arrangement details to find a suitable match.

Data flows in:

- From external entity **Student s**: *Request Information* which includes

    1. Role: driver/passenger/both

    2. Date & Time Frame xx:xx-xx:xx

    3. Starting place & Destination

    4. Party number

    5. Contact number

- From **data store D**: *Existing Request* which is the *Request Information* from another user that was stored in the database.

Process 2 compares the *Request Information* with *Existing Request*:

1. Role: Ensure there is at least one driver.

2. Date & Time Frame (xx:xx-xx:xx) : Basic validity check.

3. Starting Place & Destination: Basic validity check.

4. Party number: Ensure the total number of all the parties is less than five.

5. Contact number: Basic validity check.

Data flows out:

- To **data store D**: The user's general information, request details of potential matches

    1. Date & Time Frame that is suitable for both users

    2. Starting place & Destination

    3. Party number

    4. Contact number

Process 3 takes a user's ID that **Student s** selects and search for that ID in the database. Once it has found that ID, it retrieves *Filtered Student Info*, which is all information of the user with that ID. It will send that information to **Student s**.

Data flows in:

- From external entity **Student s** and **Data Store D:** The user's ID that **Student s** selected. It takes that ID and sent to **Data Store D** to retrieve the full personal details (*Filtered Student Info*).

Data flows out:

- To external entity **Student s**: Information that contains filtered users information (role, timeframe, date, time that would suit all users within a potential match)
- To **data store D**: After the user selects a person to see detailed information, the ID of that selected person flows out to storage D, which is for retrieving the full personal details.

Appendices

#9 Meeting with Convenors: The software development team and the convenors discussed the SRS Document, particularly how it should be laid out and what kind of details needed to be included. The team reported their current progress of the project and their plans for the design phase and beyond.

No.	Date & Time	Duration	Attendees	Agenda/Note
9	17th Aug 2020 11:00am	13 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good *Lecturers*	• Feedback on SRS • Discuss our current progress and goals • Next step is design
10	19th Aug 2020 1:00pm	19 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good *Rohesia*	• Go over SRS
11	9th Sept 2020 1:00pm	23 minutes	Aaron Win Ryan Good *Rohesia*	• Discussed our design plan with a diagram • Timeframe needs to be better explained as to what it is/means
12	12th Sept 2020 10:10am	5 ½ hours	Victor Yao Aaron Win Lysa Phan Ryan Good	• Work on diagrams
13	12th Sept 2020 7:00pm	2 hours	Victor Yao Aaron Win Lysa Phan Ryan Good	• Continue working on diagrams
14	13th Sept 2020 12:00pm	2 ½ hours	Victor Yao Aaron Win Lysa Phan Ryan Good	• Continue working on diagrams • Write descriptions for these