Southern New Hampshire University

DriverPass System Design Technical Specifications

Prepared for: Dr. Phillip Davis

Prepared by: Alexander Ahmann

Prepared on: October 17, 2024

Technical Requirements

Using the UML diagrams and DriverPass interview as a basis, I have worked out the following technical requirements needed for the development and deployment of the web application:

- A cloud-based computing environment, with the DevOps-laden CI/CD interface.
- A Linux, Apache, MySQL and PHP (LAMP) stack for the web application. The LAMP stack is the simplest of all frameworks used in web service deployment.
- An automated system for backing up, updating, and testing the security of the DriverPass service.
    - A web application firewall and cloud delivery network like CloudFlare or Impervia is a good security tool to have at our disposal.
    - The (partial) outsourcing of web application testing to independent hackers on HackerOne or BugCrowd is an idea to ponder.

UML Diagrams

Note that most of these diagrams were made with the *yEd Graph Editor*. The only exception is Figure 3, the UML sequence diagram, which was made with *Lucidchart*.

UML Use Case Diagram (Figures 1a, 1b)

Figures 1a and 1b depict two use case diagrams describing the intent of how the DriverPass service is to be used, and how its system is modelled. Figure 1a gives a perepctive of part of the greater design of the DriverPass service, and figure 1b shows a specific use case of a student creating a user account on the DriverPass service.

Figure 1a: Excerpt of DriverPass Web Application Schematic

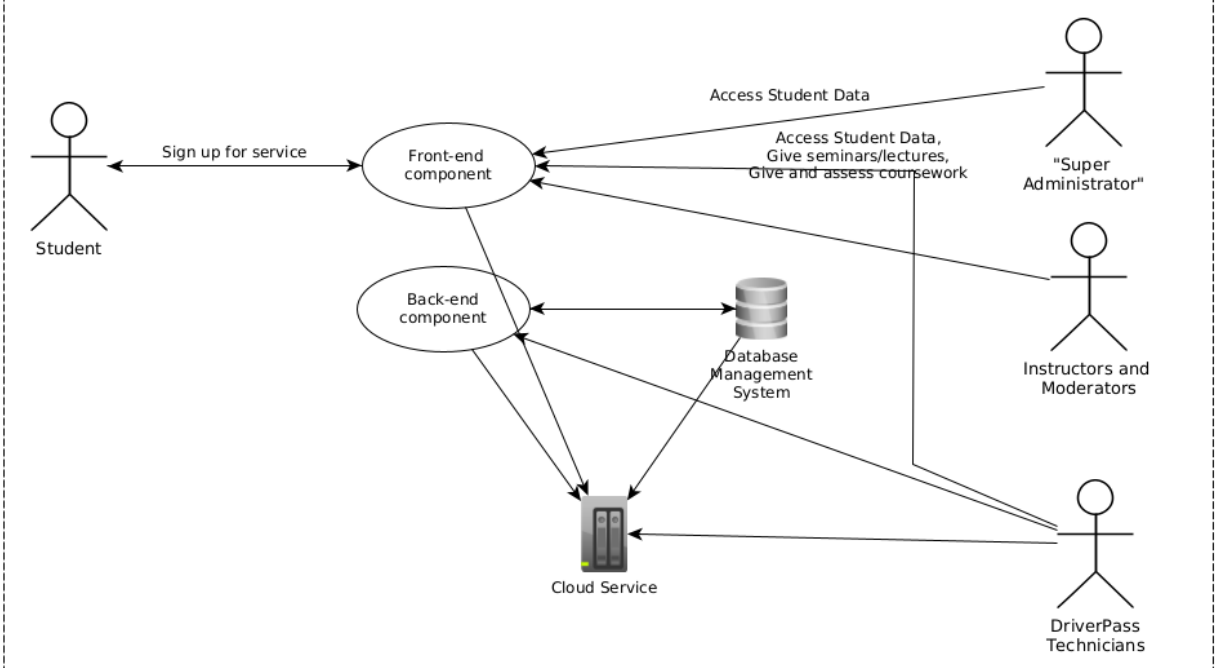
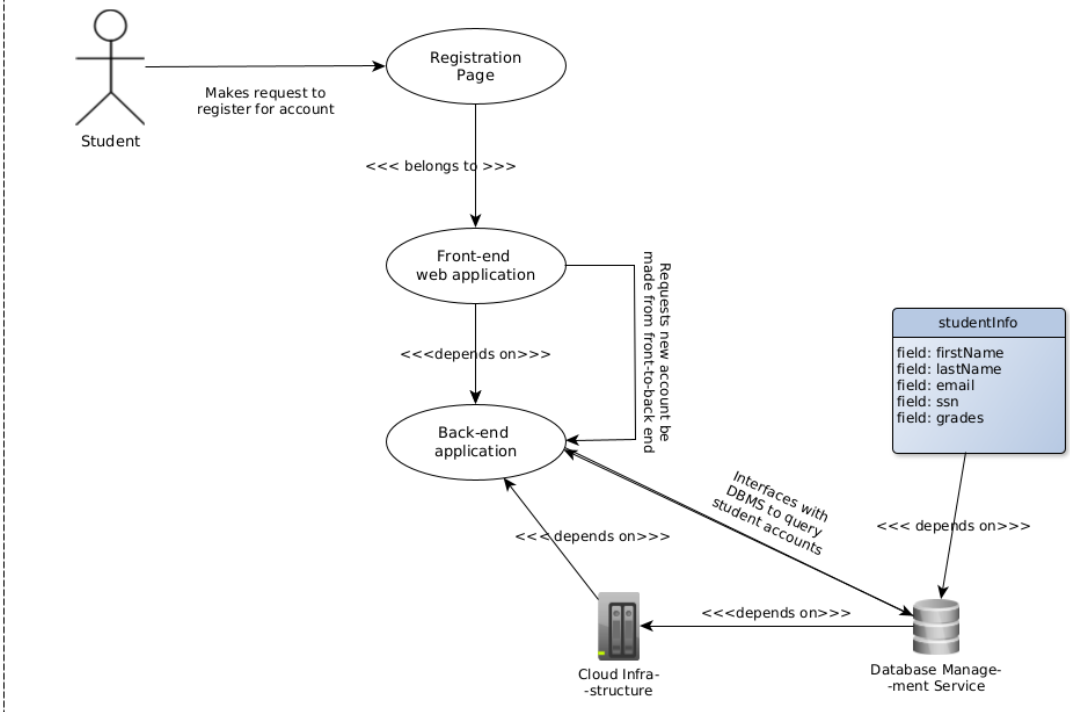


Figure 1b: Specific use case of creating a student account



UML Activity Diagrams (Figure 2a, 2b)

Figures 2a and 2b elaborate further on how the use cases work through their respective activity diagrams represented in a flowchart. In particular:

- Figure 2a depicts a student signing up for a service:
    - Step 1: Student requests a registration page.
    - Step 2: Student fills out registration form.
    - Step 3: After the student fills out the registration form, the front-end component will process the information, and then send it to the back-end component.
    - Step 4: The back-end component will interface with the database management system to get all the students registered.
    - Step 5: The back-end component will go through all the registered students and determine if the student is already registered based on the input data.
        - If the student is already registered, then the back-end component will instruct the front-end component to inform the student that they are already registered and cease the process.
        - If the student is not registered, then the procedure will move on to step 6.
    - Step 6: The back-end service will insert a record of the information supplied by the student into the SQL database.
    - Step 7: The back-end component will send an SMS code to the student, and then instruct the front-end to redirect to a page containing an input box to get the SMS code. If the student correctly inputs the SMS code, then they will be redirected to their user page. If the student does not correctly input the SMS code, then they will be sent another SMS code, and redirected to another page to get their SMS code. The process will repeat itself until the student enters the correct SMS code.
- Figure 2b depicts a super administrator requesting download of student data:
    - Step 1: The super administrator will go to the admin login page.
    - Step 2: The login page will determine whether they have an authentication cookie, and if they do, it will automatically redirect them to their user page, where they will fill out a form that will allow them to download student data.
    - Step 3: After the super administrator makes the request to get student data, the front-end web application will instruct the back-end to make a query to the database management system, and then get the dataset.
    - Step 4: The back-end component will tabulate the results, and send it to the super administrator via the front end of the *DriverPass* service.

Figure 2a: DriverPass student signup Activity Diagram

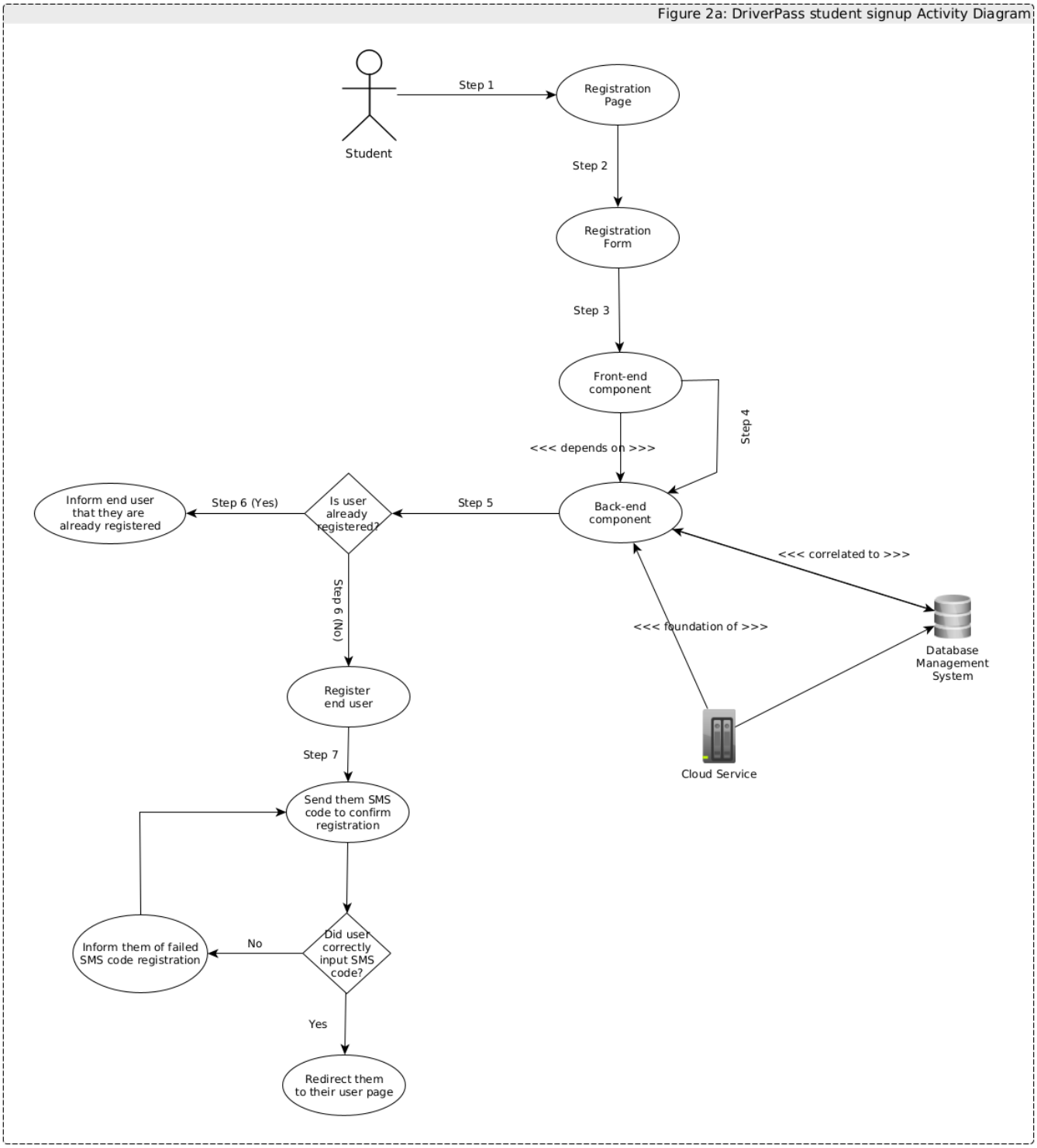
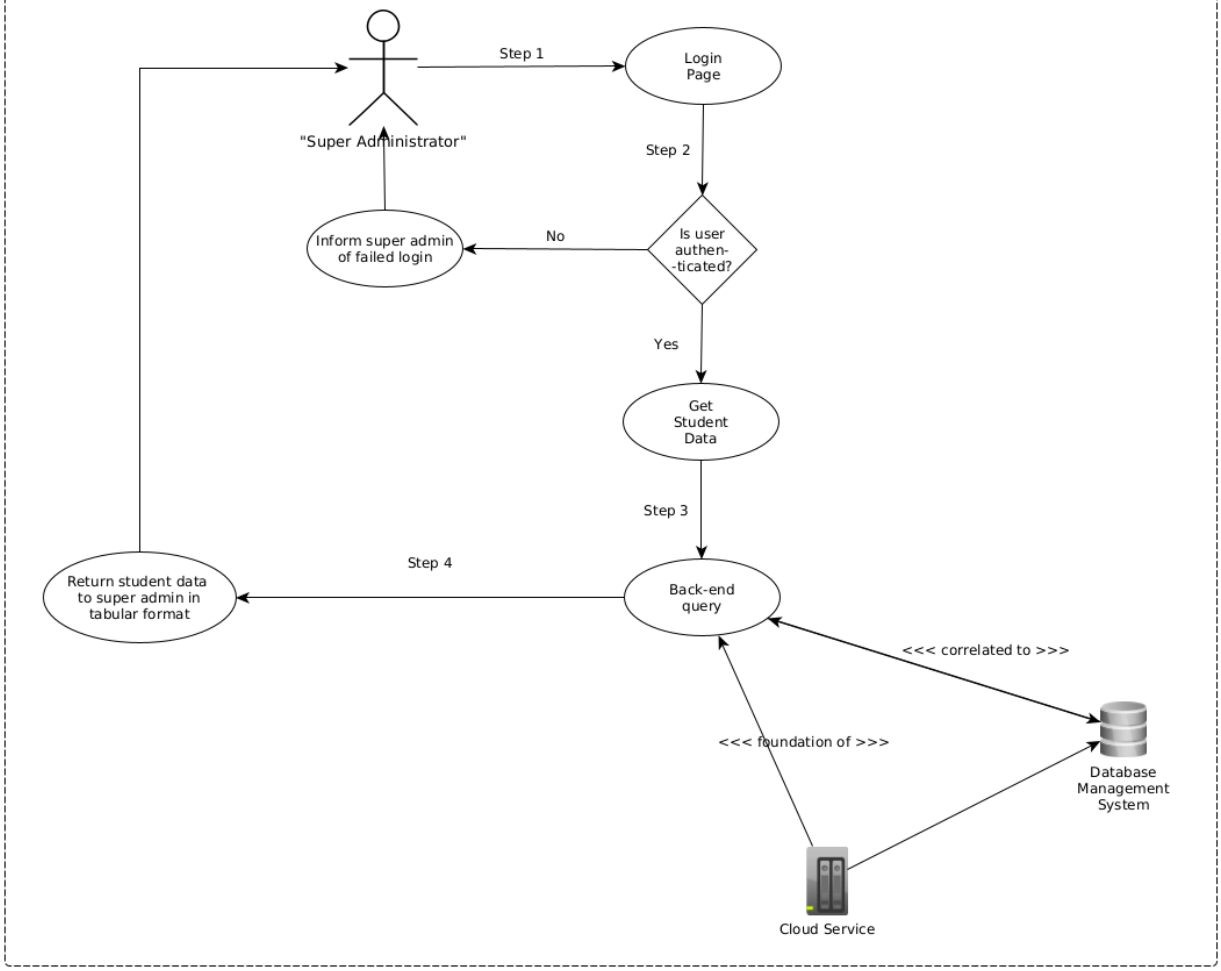
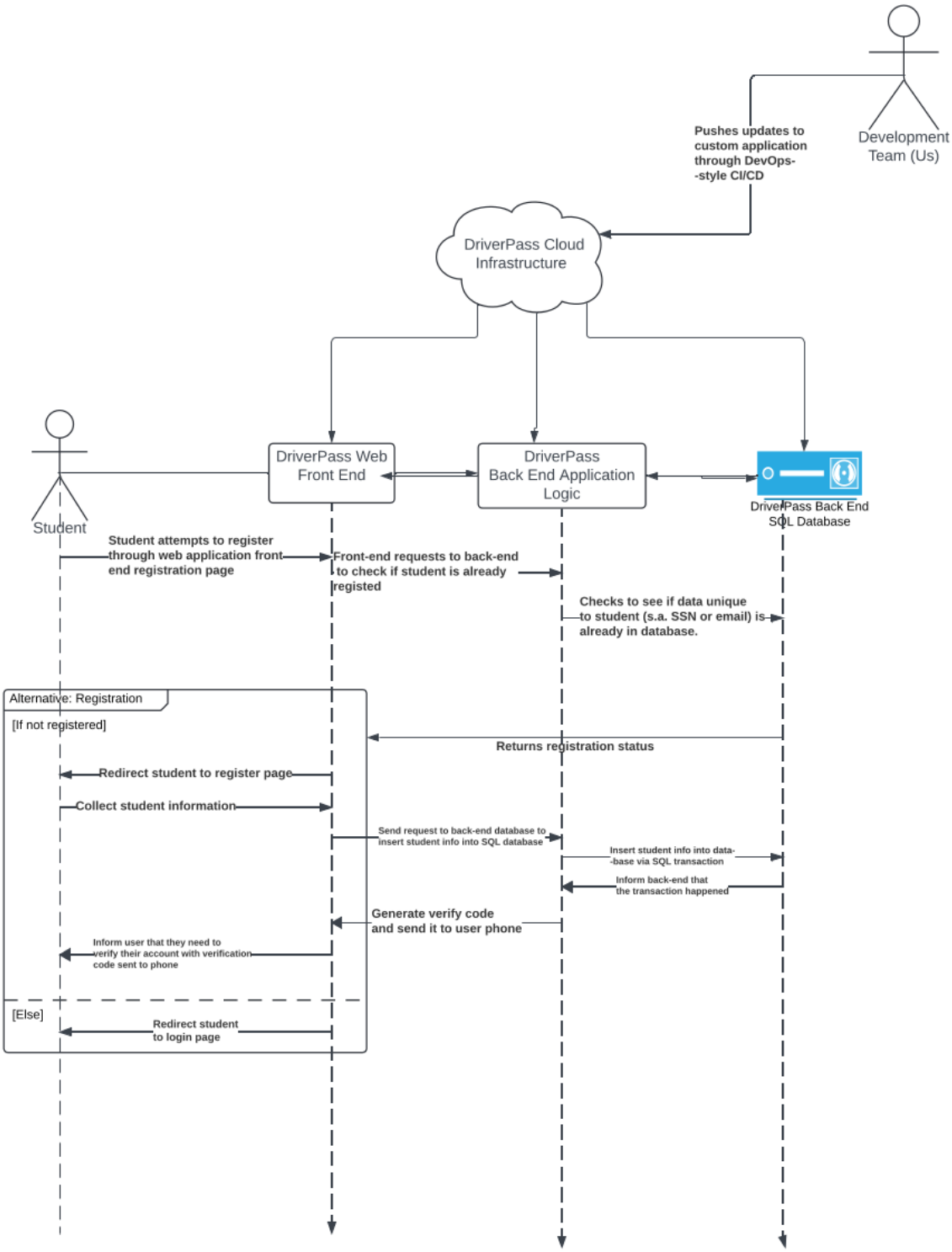


Figure 2b: DriverPass Super Administrator Request for Student Data Activity Diagram



UML Sequence Diagram (Figure 3)



UML Class Diagram (Figure 4)