Deliverable 2

Team Name: SmartGate

Project Name: Automated Barrier Gate System

Group Number: 2

Team Members: Richard Pancham (student ID: n01373454)

Jaspreet Heer (student ID: n01315290)

Name	ID	Signature	Effort
Richard Pancham	n01373454		90%
Jaspreet Heer	n01315290		90%

Project Scope

Our project focuses on creating an automated barrier gate system for parking lots. We aim to design and implement a system that streamlines the process of reserving parking spots and ensures smooth entry and exit for vehicles.

Project Goals

1. Develop a user-friendly mobile application that allows users to easily reserve parking spots based on their preferences.
2. Incorporate advanced sensors and microcontrollers to accurately detect the presence of vehicles in the parking area.
3. Create a seamless experience by controlling the barrier gate system to allow authorized vehicles to enter and exit smoothly.
4. Establish a secure and reliable database to store all the reservation information securely.
5. Prioritize user convenience by designing intuitive interfaces and enabling hassle-free payment processing for parking reservations.

GitHub Repo link and strategy

https://github.com/RichardPancham/AutomatedBarrierGate

Our Strategy for using Github is to push all of our code to the master. Given that there are only the 2 of us, we will keep up consistent communication to ensure we work on separate files at any given time and push/pull accordingly.

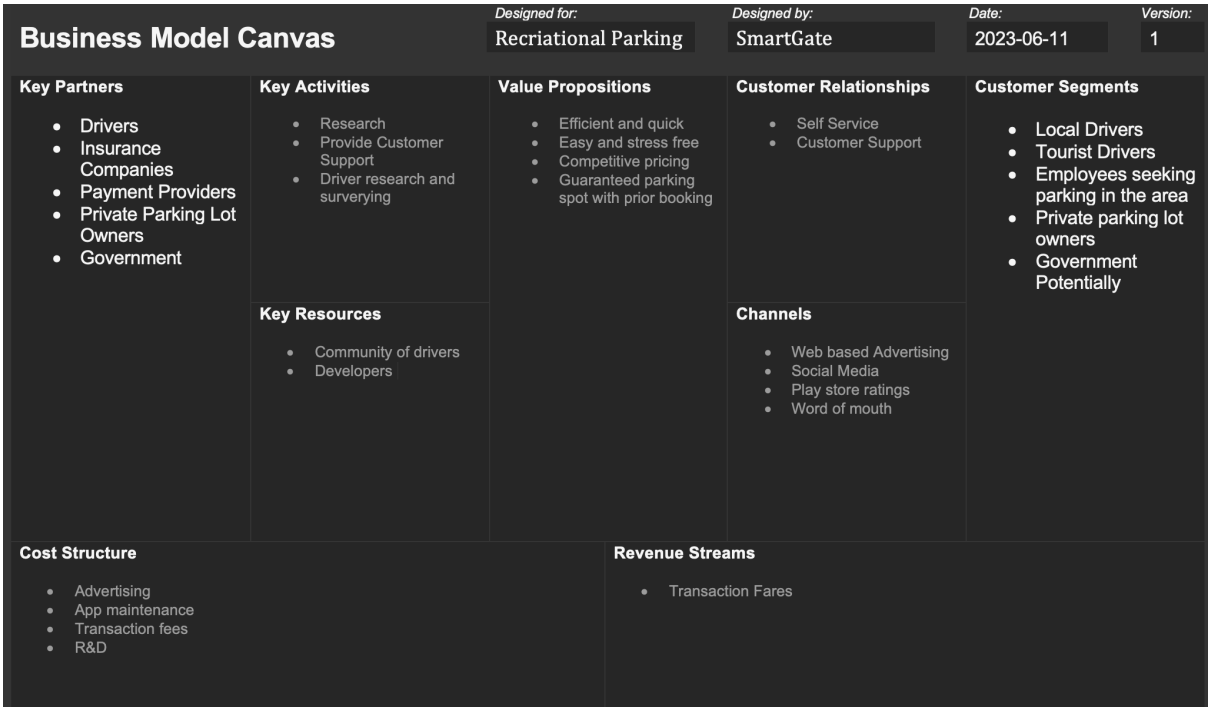
DoD Criteria:

Completed Tasks (For stories):

- UI Skeleton
-

Our DoD criteria includes having all functional components of the program implemented and tested. Code needs to be concise/efficient, and error/crash free. Usability/customer experience are also vital for our DoD requirements.

Business Model Canva:

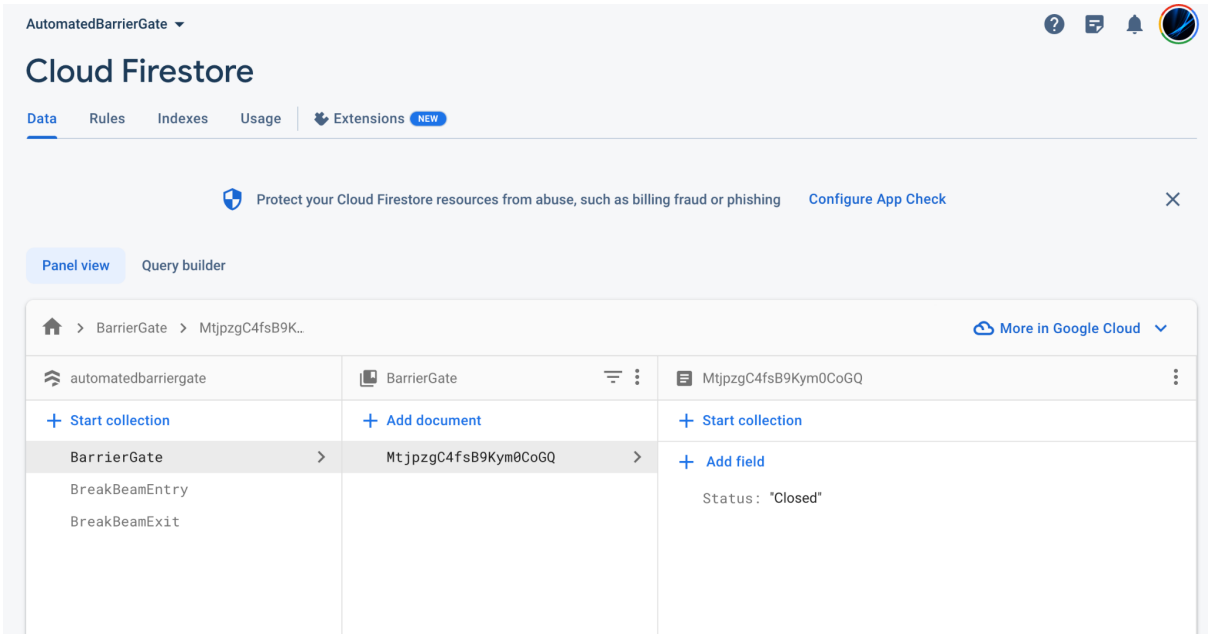


Gantt chart:

Gantt chart:

Database On Cloud:

Screenshot:



For our project, we created our database on firebase. Given our project isn't for profit, the free nature of the firebase suits our needs. The performance and reliability of this cloud solution meets our spec. Both the hardware as well as the app will utilise the database to read/writer values. The utilisation of the cloud database specifically the read and write can be viewed in the diagram below:

Software	Cloud DataBase	Hardware
	←	Breakbeam Sensor 1 (entry)
	←	Breakbeam Sensor 1 (Exit)
Barrier Open Signal	→	
	←	Luminosity Sensor

Coding Progress:

Since Deliverable 1, we have added the following to the application:

- Skeleton UI for all screens
- Splash screen
- Database integration
- Payment screen
- Selection of parking spot screen
- open/close barrier screen
- Action bar menu
- Login screen

Daily Standup:

Date:	Outcome
2023-06-02	Went into detail about the project plan and integration of sensors.
2023-06-08	Finalised the skeleton of the app and database integration.
2023-06-10	Checked in on progress for assigned tasks and went over some intricacies of the application which were discovered.

6 stories and each story must be split to minimum of 5 tasks

Story 1: User Registration and Login

Tasks:

1. *Design a splash screen with your app logo.*
2. *Create a login screen where users can enter their emailand password.*
3. *Implement Firebase Authentication for secure user registration and login.*
4. *Display informative error messages for invalid login attempts.*
5. *Utilize Firebase Authentication to store user login information securely.*

Story 2: Parking Selection

Tasks:

1. *Design a interface with a tab for parking selection.*
2. *Connect the app to Firebase Realtime Database.*
3. *Fetch parking data from Firebase and display it in the app.*
4. *Allow users to choose a parking spot from the available options.*
5. *Update Firebase to mark the selected parking spot as reserved.*

Story 3: Payment for Parking Reservation

Tasks:

1. *Create a tab within the app for payment processing.*
2. *Integrate a reliable payment gateway for secure transactions.*
3. *Calculate the total cost of the parking reservation.*
4. *Handle payment processing using the chosen payment gateway.*
5. *Update Firebase to store the payment details associated with the parking reservation.*

Story 4: Time Selection for Parking Reservation

Tasks:

1. *Implement a tab within the app for selecting the reservation time.*
2. *Design a user-friendly time picker or calendar view for selecting start and end times.*
3. *Validate the chosen time to ensure it meets the requirements.*
4. *Calculate the duration of the parking reservation based on the selected time.*
5. *Update Firebase to store the reservation time associated with the parking spot.*

Story 5: Gate Control

Tasks:

1. *Add a button within the app for gate control.*
2. *Implement the necessary logic to control the parking gate.*
3. *Integrate the app with the servo motor to enable gate movement.*
4. *Handle user interactions with the gate control button, providing feedback on gate status.*
5. *Ensure smooth gate operation through the app.*

Story 6: App Menu and Additional Features

Tasks:

1. *Create a menu option within the app for accessing additional features and settings.*
2. *Design a settings screen for users to customize app preferences.*
3. *Implement a help dialer for users to seek assistance if needed.*
4. *Include an option to open a specific website within the app.*
5. *Ensure user-friendly navigation and functionality for menu options and additional features.*

Trello link : https://trello.com/b/P47hMzsG/automated-barrier-gate