**Deliverable: Design**

**What will the API do and how will it work?**

The API that I am going to create will work by connecting the client and the server, by sending specific http requests depending on what the client’s needs are. The API must allow users to top up their employee cards from the kiosk terminals. This means that it must communicate requests from the kiosk terminals to the server that deals with these requests. For example, a user wanting to check their balance would send a GET request to the server and in return receive their balance but if they wanted to top-up their balance by £20 they would send a PUT request for +20.

**Assumptions:**

* The users know how to use the kiosks
* The users can read
* There are different levels of staff, for example ‘workers’ and ‘management’
* Management have authority over workers and can view a list of all employees from their account
* Every card already exists in the database, it just might not be assigned to an employee yet. For example, there could be 10 cards registered in the database, but only 5 employees working for the company.
* Cards are added manually to the database, users will not be manually added and these will be done via the kiosk

**Changes or additions made to the requirements:**

When creating my API, I would like to include every type of HTTP request, therefore I will be introducing ‘management’ and ‘workers’. Management have the authority to tap their card and delete users if they deem necessary. For example, someone gets fired and their card would no longer be needed. This is when management would then tap their card, the system would read their employee information and understand that they are a manager. From here the management can create DELETE requests to delete specific user ID’s from the system (the employee who just got fired).

Management could also use their card to create new users. And updating the database of current employees working for Bows Formula One High Performance Cars requires creating POST requests. A POST request is used to add users.

Above I have covered how the API will use GET, POST, PUT and DELETE http requests.

**Use Cases:**

**A close up of a logo

Description automatically generated**

A screenshot of a cell phone

Description automatically generated**Sequence Diagrams:**

**Data Model:**

There are only 2 tables in the database. They have a one – one relationship.

Here is how they are linked:

**Entity Relationship Diagram:**

|  |
| --- |
| **Employee** |
| ID (PK) |
| Name |
| Email |
| Mobile |
| Pin |
| Session Time (NULL WHEN REGISTERING) |

|  |
| --- |
| **Card** |
| ID (PK) |
| Employee ID (FK) |
| Balance |