**Test Plan:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Number** | **Test Description** | **Test Tools** | **Expected Outcome** | **Actual Outcome** | **Pass / Fail** | **Further Work** |
| **1** | Can the user log in with the correct card ID and pin? | Postman  iTerm2 (to run the application) | The user should receive a welcome message when entering correct login credentials | A welcome message appears | Pass | - |
| **2** | Can the user log in with an incorrect pin? | Postman  iTerm2 (to run the application) | The user should receive a message informing them that they are using an incorrect pin | A message with “incorrect pin” appears | Pass | - |
| **3** | Can the user log in with a card that doesn’t exist in the database? | Postman  iTerm2 (to run the application)  MySQLWorkBench | The system should print out “Card not found” | The system prints out “Card not found” | Pass | - |
| **4** | Can the user log out when tapping their card a second time? | Postman  iTerm2 (to run the application) | The system should print “Goodbye <name>” after tapping for a second time | The system prints “Goodbye <name>” after tapping for a second time | Pass | - |
| **5** | Can the user view their balance? | Postman  iTerm2 (to run the application)  MySQLWorkBench | The user’s balance should appear from the database | The user’s balance prints on the screen. It matches the database balance | Pass | - |
| **6** | Can the user top up their balance? | Postman  iTerm2 (to run the application) | The user should be able to type out the amount they want to increase their balance by | The user’s balance is updated with the amount specified | Pass | - |
| **7** | Does the session expire after 2 minutes of inactivity? | Postman  iTerm2 (to run the application) | The user should not be able to access any services and the system will print session expired and ask them to log in again | The user is asked to log in again. They cannot view any services | Pass | - |
| **8** | Can the user top up their account with invalid information? For example adding a string to their balance rather than a double? | Postman  iTerm2 (to run the application) | The user should not be able to input anything other than an integer or double | The system prints a 400 Bad Request and says it cannot accept a string | Pass | - |
| **9** | Can a user use the kiosk services without logging in or after tapping twice and logging out? | Postman  iTerm2 (to run the application) | The system should print out a message saying their session has expired | The system prints out a messaging saying that the session has expired | Pass | - |
| **10** | Can a user use the kiosk services after the session has expired after 2 minutes of inactivity? | Postman  iTerm2 (to run the application) | The system should print out a message saying their session has expired | The system prints out a messaging saying that the session has expired | Pass | - |
| **11** | Can a user register a new card with a pin longer than 4 digits? | Postman  iTerm2 (to run the application) | User requirements ask for a 4 digit pin however I haven’t added this part into the code so any length should be accepted | The user can create a pin of any length | Does what is expected however this goes against the business requirements asking for a “four digit pin number” | A check for pin length needs to be added and make sure it prevents users from creating a pin shorter or longer than 4 digits |
| **12** | Can a new user register after filling out all fields? | Postman  iTerm2 (to run the application)  MySQLWorkbench | New users should be able to register after filling out necessary fields | A new user can register and the database will be updated | Pass | There is no check that each field is populated |
| **13** | Can a user register without filling out all fields? | Postman  iTerm2 (to run the application)  MySWLWorkbench | The system will fail with a 400 Bad Request since not all fields are populated | A user can register without filling in all fields and the database field will be populated with “Null” instead | Fail | A check to ensure all fields are populated needs to be implemented |
| **14** | Can a manager view all employees currently registered with the system? | Postman  iTerm2 (to run the application) | The system should print out all information on each employee | The system prints out, in JSON format, a list of all employees with information about them | Pass | - |
| **15** | Can a manager delete an employee if they were no longer working at the company? | Postman  iTerm2 (to run the application)  MySQLWorkbench | The system will print out “user deleted” after the manager specifies the employee ID to delete | The employee specified gets deleted from the database and the system prints out “user deleted” | Pass | - |

**Evidence:**

|  |  |
| --- | --- |
| **Test Number** | **Evidence** |
| **1** | **A screenshot of a social media post  Description automatically generated** |
| **2** | **A screenshot of a social media post  Description automatically generated** |
| **3** | **A screenshot of a social media post  Description automatically generated** |
| **4** | **A screenshot of a social media post  Description automatically generated** |
| **5** | **A screenshot of a social media post  Description automatically generated** |
| **6** | **A screenshot of a social media post  Description automatically generated** |
| **7** | **A screenshot of a social media post  Description automatically generated** |
| **8** | **A screenshot of a social media post  Description automatically generated** |
| **9** | **A screenshot of a social media post  Description automatically generated** |
| **10** | **A screenshot of a social media post  Description automatically generated** |
| **11** | **A screenshot of a social media post  Description automatically generated** |
| **12** | **A screenshot of a social media post  Description automatically generatedA screenshot of a cell phone  Description automatically generated** |
| **13** | **A screenshot of a cell phone  Description automatically generated** |
| **14** | **A screenshot of a social media post  Description automatically generated** |
| **15** | **A screenshot of a social media post  Description automatically generated** |

**Unit Testing:**

I did not manage to do unit testing, however if I had the time I would have created some test cases for each unit of my work and reviewed these/improved my current code if necessary. I would have used Junit as my testing tool.

**Integration Testing:**

I completed integration testing, as explained in my evaluation and also evidence in the test plans above. I have several units that were required to work together to complete all of the business requirements. For example, the user cannot view their balance if they have not logged in. The user cannot top-up their balance if their session has expired. If a user has been deleted from the database then they can not log in again without registering. All of these individual units were tested and they were tested against eachother. For example originally I did have it so that a user could just view their balance if they put the correct parameters in the JSON text but they did not have to be logged in. This was because it was a separate module and I wanted to just be able to test this individually first. After completing each module I integrated the units together and made sure that the system worked as requested once everything was merged together.

**System Testing:**

I completed full end to end testing, evidenced in the test plans above. I had a user log-in, view his balance and then top up his balance. I confirmed this was all correct in the database after by using MySQLWorkbench (GUI to view a database). I also waited 2 minutes to test that the timeout feature worked and after the 2 minutes I then tested the view balance endpoint and it did not work because the session had expired. Upon logging in again this did work, as expected. After completing a full end to end test I finished it off with the login endpoint again and logged out. Again, no services worked after logging out and required the user to log back in.

I also tested the manager level requirements, not part of the business requirements but instead additional features I wanted to add to help cover my use of HTTP requests. The view all employees endpoints work as expected as well as the ability to delete a user based on the ID specified.

**Acceptance Testing:**

I have met every single requirement under the ‘Busniess Requirements’ section of the brief. I have:

* Created a RESTful web service which allows users to use their existing data cards to register and top up with money
* Cards that are not registered on the system will asked to be registered with the following information filled out:

Employee ID, Name, Email and Mobile Number

* A pin is chosen by the employee upon registering
* The data for all of this is held and stored in a database
* If a card is not registered then it will respond with a messaging saying ‘Card not found’ and will then have to register
* When a user taps their card again (calls the login endpoint twice) it will log the user out and print “Goodbye”
* The application times out after 2 minutes of inactivity