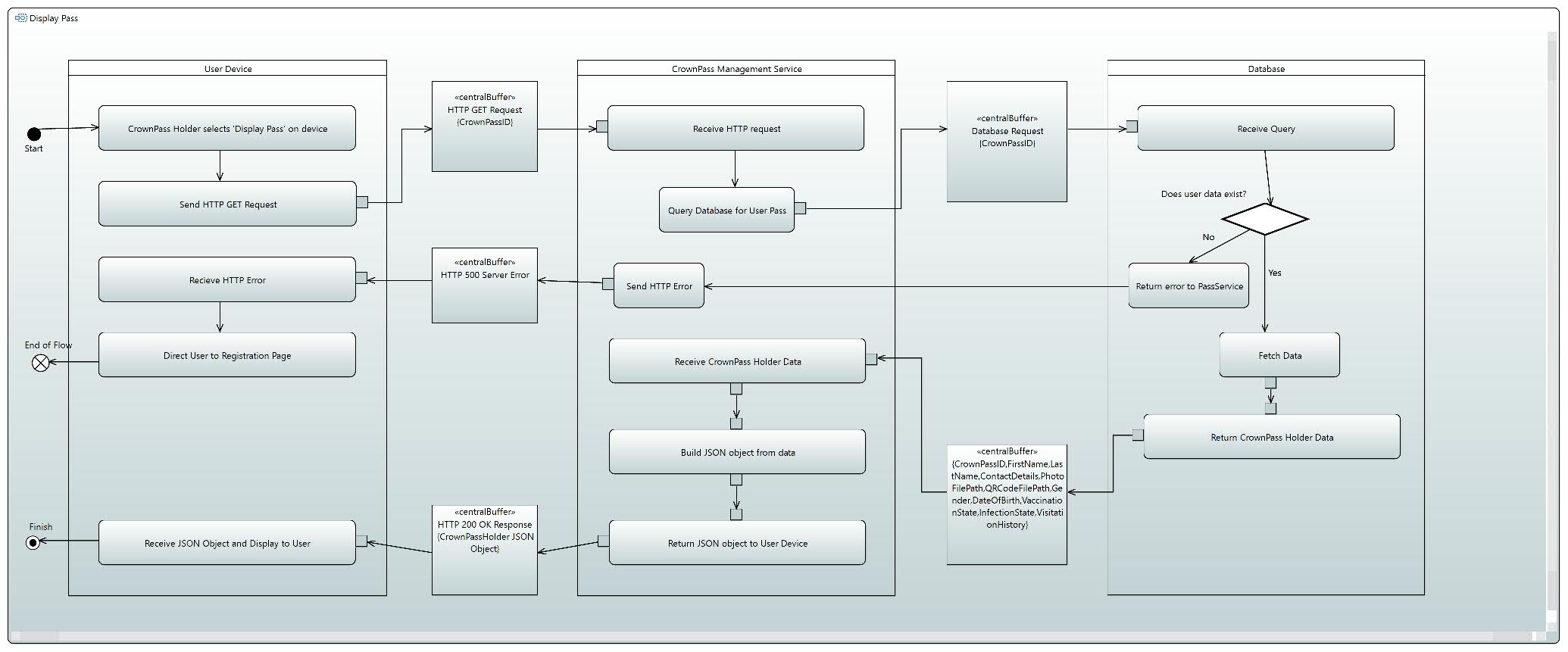
**COMP6030 – Software Testing**

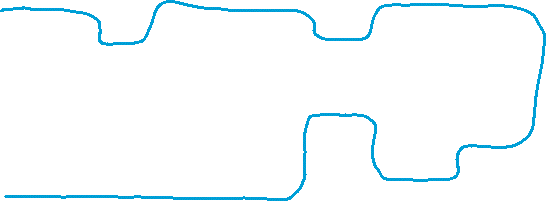
Unit Test Plan

Class to test: CrownPassHolder

|  |  |  |
| --- | --- | --- |
| Test Case | Method and Parameters | Expected Output |
| Check in to Controlled Area | checkIn(ControlledArea) | Return newly created Visit Object. |
| Failed Check in to Controlled Area (Invalid Controlled Area) | checkIn(ControlledArea) | Error message raised stating that the Controlled Area supplied is invalid. |
| Failed to Check in to Controlled Area (Maximum Capacity Reached) | checkIn(ControlledArea) | Error message raised stating that the Controlled area is current at maximum capacity. |
| Failed to Check in to Controlled Area (User does not have Green InfectionState) | checkIn(ControlledArea) | Error message raised stating that the User does not qualify to enter Controlled Area due to their InfectionState not being green. |
| Check out of Controlled Area | checkOut(Visit) | CheckOutTime on latest Visit object in VisitationHistory is updated to the current time. |
| Failed to check out of Controlled Area (Already checked out) | checkOut(Visit) | Error message raised stating that Visit has already been checked out. |
| Failed to check out of Controlled Area (Not checked in/No Visit Exists) | checkOut(Visit) | Error message raised stating that the user does not have a Visit on record to be able to check out of. |
| Print CrownPass | printPass() | A copy of the CrownPassHolder object is sent to the printing service. |
| Unable to Print CrownPass | printPass() | Error message raised stating that the printing service is unavailable. |
| Notify CrownPassHolder | notifyUser(message:string) | Calls the Notification Service with the correct notification message. |
| Unable to Notify CrownPassHolder | notifyUser(message:string) | Error message raised stating the Notification Service is unavailable. |
| Book a vaccination | bookVaccination(VaccinationCenter,DateTime) | Calls the appointment service to create an appointment at the Vaccination Center at the specific time. Appointment Service should create a new Appointment object in user’s ‘Appointments’ property. |
| Unable to book vaccination at the provided time | bookVaccination(VaccinationCenter,DateTime) | Error message raised stating that the Appointments service was unable to book the correct time because the time was unavailable at that Vaccination Center. |
| Unable to book vaccination at the provided Vaccination Center | bookVaccination(VaccinationCenter,DateTime) | Error message raised stating that the Appointments service was unable to book because the Vaccination Center provided was invalid. |
| Display CrownPass of holder | displayPass() | Returns a copy of the CrownPassHolder object to the PassService so it can return it to the user device. |
| Unable to display CrownPass | displayPass() | Error message raised stating that the Pass service is unavailable. |
| Get ID of user | getCrownPassID() | Returns a uint64 number representing the CrownPass ID. |
| Get First Name of user | getFirstName() | Returns a string representing the first name of the CrownPass user. |
| Get Last Name of user | getLastName() | Returns a string representing the last name of the CrownPass user. |
| Get Contact Details of user | getContactDetails() | Returns a ContactDetails object containing the contact details for the CrownPass user. |
| Get file path of user’s photo | getPhotoFilePath() | Returns a string representing the file path where the user’s photo is stored on the web server. |
| Get file path of user’s QR code | getQRCodeFilePath() | Returns a string representing the file path where the user’s QR code is stored on the web server. |
| Get user’s gender | getGender() | Returns an EGender enumerable representing the user’s gender. |
| Get date of birth of user | getDateOfBirth() | Returns a DateTime representing the date of birth of the user. |
| Get vaccination state of user | getVaccinationState() | Returns an EVaccinationColor enumerable representing the vaccination state of the user. |
| Get infection state of user | getInfectionState() | Returns an EInfectionColor enumerable representing the infection state of the user. |
| Get history of visited controlled areas for user | getVisitationHistory() | Returns a list of Visit objects representing the history of Visits that a user has made to Controlled Areas. |
| Get user’s appointments | getAppointments() | Returns a list of Appointments representing the current appointments that the user has booked. |
| Set user’s name | setName(FirstName,LastName) | CrownPassHolder FirstName and LastName properties are updated to the new values. |
| Set user’s contact details | setContactDetails(ContactDetails) | CrownPassHolder’s ContactDetails field are updated to be the same as the ContactDetails object provided. |
| Set file path for user’s photo | setPhotoFilePath(string) | CrownPassHolder’s PhotoFilePath field is updated to the value provided. |
| Set file path for user’s QR code | setQRCodeFilePath(string) | CrownPassHolder’s QRCodeFilePath field is updated to the value provided. |
| Set user’s gender | setGender(EGender) | CrownPassHolder’s Gender field is updated to the value provided. |
| Set user’s date of birth | setDateOfBirth(DateTime) | CrownPassHolder’s DateOfBirth field is updated to the value provided. |
| Set user’s vaccination state | setVaccinationState(EVaccinationColor) | CrownPassHolder’s VaccinationState field is updated to the value provided. |
| Set user’s infection state | setInfectionState(EInfectionColor) | CrownPassHolder’s InfectionState field is updated to the value provided. |
| Authenticate user | logIn(password:string) | User receives authentication token from Authentication Service. |
| Authenticate user fails due to incorrect password. | logIn(password:string) | User receives an HTTP 401 response from the Authentication Service stating that the password was incorrect. |
| Authenticate user fails due to invalid user. | logIn(password:string) | User receives an HTTP 404 response stating that the user was not found by the Authentication Service. |
| Log out user | logOut() | Authentication Service invalidates user’s authentication token. |
| Log out user fails (user is not logged in) | logOut() | Authentication Service sends error message stating that user is not currently logged in. |

System Test Plan





Scenario 1 | Scenario 2

**Use Case: Display Pass**

Scenario 1: Successful request from user to display pass on their device.

Description of Scenario 1:

|  |  |  |
| --- | --- | --- |
| User | CrownPassHolder MGMT Service | Database |
| 1. Authenticated user selects ‘Display Pass’ on their device. |  |  |
| 2. Device sends HTTP GET request to web server. | 3. CrownPass service receives request to get CrownPass for specified user. |  |
|  | 4. CrownPass service sends query to database to check CrownPassHolder data exists. | 5. Database receives query. |
|  | 7. CrownPass service receives data for CrownPassHolder from database. | 6. Database checks data exists. |
| 9. User receives HTTP OK Response with CrownPass data. | 8. CrownPass service builds a JSON object from received data to return in HTTP response payload. |  |
| 10. Mobile app formats data to display to user. |  |  |

Scenario 2: User data does not exist in database (Registration Incomplete).

Description of Scenario 2:

|  |  |  |
| --- | --- | --- |
| User | CrownPassHolder MGMT Service | Database |
| 1. Authenticated user selects ‘Display Pass’ on their device. |  |  |
| 2. Device send HTTP GET request to web server. | 3. CrownPass service receives request to get CrownPass for specified user. |  |
|  | 4. CrownPass service send query to database to check CrownPassHolder data exists. | 5. Database receives query. |
|  | 7. CrownPass service receives error from database stating that data does not exist for user. | 6. Database checks data exists. |
| 9. User receives HTTP error response. | 8. CrownPass service sends HTTP error response. |  |
| 10. Mobile app redirects user to registration service to complete registration. |  |  |

**Test Case for Scenario 1:**

*Test Data:*

Input:

* CrownPassID

Stored Data:

* On Mobile Device: CrownPass ID.
* On Database: User’s first name, last name, photo URL, QR Code URL, contact details, gender, date of birth, vaccination state, infection state, visitation history and upcoming appointments.

Output:

* JSON Object containing user’s first name, last name, photo URL, QR Code URL, contact details, gender, date of birth, vaccination state and infection state.

*Test Process:*

1. Set up test.

a. Create CrownPassHolder object in database and ensure all fields are populated.

b. Ensure user is authenticated before running test.

2. Select ‘Display Pass’ on device.

a. Check HTTP GET request is sent to the correct URL of the API endpoint.

3. Check request is received at the CrownPass Management Service (redirected via API endpoint gateway).

a. Expected Output: HTTP GET request contains a payload including the User’s CrownPassID

4. Check that the database receives a query from the CrownPass Management Service

5. Check that database returns the correct CrownPass data for the provided CrownPassID.

a. Expected Output: Database returns correct CrownPassHolder object to CrownPass Management Service.

6. Check that CrownPass Management Service builds a JSON object using fields from the CrownPassHolder object.

a. Expected Output: JSON object containing correct information about the CrownPassHolder.

7. Check that the HTTP response received by the user’s device contains the JSON object.

8. Check that the mobile device receives the JSON object.

a. Expected Output: HTTP 200 OK response with JSON object containing CrownPassHolder data.

9. Ensure that the mobile device can download the Photo and QR codes using the URL provided in the JSON object.

a. Expected Output: Photo and QR code are successfully downloaded to the device.

**Test Case for Scenario 2:**

*Test Data:*

Input:

* CrownPassID

Stored Data:

* On Mobile Device: CrownPass ID.
* On Database: User’s first name, last name.

Output:

* HTTP 500 Error Response

*Test Process:*

1. Set up test.

a. Create CrownPassHolder object in database and ensure only first name and last name fields are populated.

b. Ensure user is authenticated before running test.

2. Select ‘Display Pass’ on device.

a. Check HTTP GET request is sent to the correct URL of the API endpoint.

3. Check request is received at the CrownPass Management Service (redirected via API endpoint gateway).

a. Expected Output: HTTP GET request contains a payload including the User’s CrownPassID

4. Check that the database receives a query from the CrownPass Management Service.

5. Ensure that database returns an error message to the CrownPass Management Service.

a. Expected Output: Error message from the database received by CrownPass Management Service stating that the data for the specified CrownPassID does not exist.

6. Ensure that the CrownPass Management Service returns an HTTP 500 error response to the user device stating that the data does not exist for the specified CrownPassID.