Mail Database Project

Christian Johnson & Dan Nusraty & Dylan McGill
Optimail

April 1, 2024

Contents

1 Preface

Tasks taken to complete this assignment:

- Compiled all code files and class documents to Git repo
- Went over code structure and general function of the system in its current state
- Discussed how best to test various components of the application
- Tested several components of the system
- Fixed issues with application
- Pushed working, fixed code to Github

Changelog:

- Tables are more visually appealing
- Added a logout button (currently non-functional)
- Added USCGA logo
- Tables now resize automatically (will shrink if empty, also allows app to fullscreen)
- Added search capability to Data tab
- Added Delete capability to Data tab
- Added edit button to Data tab, now raises submenu when clicked (main functionality still WIP)

2 Test Plan

2.1 Unit Testing

Note: These tests will have to be performed manually, as our app is primarily GUI focused.

Login:

- This test verifies the behavior of the login screen.

- It checks if the screen redirects the user to the Home screen upon successful login with valid credentials.
- Since the app is GUI-focused, this test would involve manually entering valid credentials and observing the redirection to the Home screen.

Data Entry:

- This test ensures that the "Add Package" button correctly records data.
- It specifically checks if the tracking number is recorded accurately and if the package is associated with the correct cadet.
- Manual interaction with the GUI would be required to input data and verify the recorded information.

Search Function:

- This test validates the search functionality.
- It verifies that a search returns all valid results or, alternatively, that it does not return any invalid results.
- Again, manual interaction with the GUI is necessary to perform searches and confirm the correctness of the results.

2.2 Integration Testing

Database Integration

- Verify that the data entered through the GUI is correctly stored in the database.
- Test if package information, including tracking numbers and associated cadet details, is accurately saved in the database after clicking the "Add Package" button.
- Ensure that the search functionality retrieves the correct data from the database.

Email Notification Integration

- Test if the email notification feature functions as expected.
- Verify that when a package is added and associated with a cadet, an email notification is sent to the corresponding cadet's email address.
- Check if the content of the email notification contains accurate package information.

User Authentication Integration

- Ensure that the login screen interacts correctly with the authentication system.
- Test if valid credentials provided on the login screen authenticate the user and grant access to the application's features.
- Verify that invalid credentials result in appropriate error messages or prevent access to sensitive functionalities.

GUI Navigation Integration

- Test the navigation flow between different screens or tabs of the GUI.
- Verify that users can seamlessly switch between tabs/screens without encountering errors or unexpected behavior.
- Ensure that data entered in one tab/screen remains consistent and accessible when navigating to another tab/screen.

Report Generation Integration

- Verify that the report generation feature utilizes the correct data from the database.
- Test if generated reports accurately reflect the package tracking data stored in the database.
- Check if generated reports can be exported/downloaded in the desired format (e.g., PDF, Excel).

2.3 Defect tracking

We plan to use the "Issues" functionality built in to Github. When a team member or test user notices an issue with the application, they can submit an Issue on Github detailing the process for reproducing the issue, as well as potential expected outcomes.

2.4 Test Schedule

Unit testing will be an ongoing process throughout the development life-cycle, ensuring the proper function of each application component as it is completed. Prior to integration testing, another round of unit testing will occur once all major functionalities are implemented. Integration testing will commence once the application reaches a usable state, which we anticipate to be approximately 1-2 weeks from now.

3 Working Prototype

Github Repo

4 Demonstration Video

Video Link
Click "View Raw" on Github to download the video file.

5 Project Management

5.1 Schedule

Next Up:

- Finalize "Manage DB" tab and polish the user interface (Effort - 4)
- Create "Settings" tab and allow for user specific configuration settings (Effort - 8)
- Implement Logout Button (Effort - 3)
- Create "Reports" tab and determine what types of reports to logically implement (Effort - 9)

Long Term:

- Implement physical scanner (Effort - 6)
- Create larger example database to full scale test (Effort - 6)
- Optimize code (Effort - 7)
- Allow file operations, to export/import/backup actual database file. (Effort - 8)

5.2 Meeting Summaries

Meeting 1

- Reviewed Previous work and compiled upcoming plan
- Worked on "Manage DB tab" in a group environment

Meeting 2

- Discussed test plans, and how to put one together.
- Began work on Phase 05.
- Added finishing touches to code in order to present the "Working prototype" for core functionality.

Meeting 3

- Worked on test plans.
- Recorded demonstration video.