IST311 (Fall 2018)

Group 2

September 4, 2018

Pool Access System

# Introduction

The Pool Access System (PAS)’s overarching goal is to eliminate the need for physical passes that provide access to a pool and physical sign in sheets to track attendance. The final product will allow pool attendants to check members in simply by looking them up. Checking in and out will automatically track attendance numbers, reducing a large amount of administrative work for attendants.

The development period is broken up into five sprints, each lasting 2 weeks. Sprint #1 focuses around developing the functionality to allow member information to be registered into the system. This includes proper error checking functionality, as well as developing proper access levels to be used.

Sprint #2 focuses on additional registration details such as updating a member’s registration. It also focuses on creating proper JSON files to support the systems member registries.

Sprint #3 focuses on creating the necessary search functionality for the system to allow attendants to look up members, and begins the process of developing the “view” or User Interface by creating proper panels to allow search results and member info to be displayed.

Sprint #4 focuses on finishing up all functionality to allow members to access the pool. This sprint will include creating the “current swimmers” section of the application and all UI functionality necessary.

Sprint #5 will focus on the proper logging of pool attendance numbers, updates to the members registration, etc. This functionality is implemented last because it relies on proper new member registration, and checking in and out of members to function correctly.

We believe that following the development strategy and schedule listed below in detail will allow PAS to be developed at a quick and consistent speed.

# User Stories

|  |  |
| --- | --- |
| User Story # | User Story |
|  | Pool operator (PO) can find a customer in the database. PO types-in a part of the user’s name, address, or a phone number into the search field. The system displays top matches from the database of registered pool customers (Search Form). The Search Form includes customer’s personal info (photo, name, age, phone) as well as information on entrance permission and other constraints imposed on the Customer (such as requires adult supervision) if any. |
|  | PO can see detailed information about one customer. The system displays the form with the detailed information about the customer (Detailed Form), including a photo, full name, date of birth, allergies, emergency contact information, and swimming ability. |
|  | PO can check-in a customer by clicking the respective button either on the Search Form or on the Detailed Form. In turn, the system logs the event and adds the customer to the list of current swimmers. |
|  | Pool Administrator (PA) can add a new customer to the database of registered pool customers. The system displays the Detailed Form. PA has to fill in all required fields and the system has to validate the form before storing the new record in the database. |
|  | PA can edit all customer-related information. The system displays the Detailed Form in the edit mode and updates the record only after successful validation. |
|  | PO (or PA) can view the current swimmers in the pool. The system displays information on all swimmers who have checked-in but have not checked-out in a special form (Status Form). |
|  | PO can check-out a single customer either on Status Form, Search Form, or Detailed Form. The system logs the event and eliminates the customer from the list of current swimmers. |
|  | In the Reports section, PA (or PO) can see the list of all Visits during the specified period of time. The system displays the report (Visits Report) according to the specified Start & End dates. |
|  | In the Reports section, PA (or PO) can see the list of all Customers during the specified period of time. The system displays the report (Customers Report) according to the specified Start & End dates. |
|  | In the Reports section, PA (or PO) can see the attendance statistics for the specified period of time. The system displays the report (Attendance Report) according to the specified Start & End dates. PA has to select the statistic he is interested in (minimum, average, or maximum load; total count). The system shows the report as a table, where columns represent days of the week, and rows represent hours of the day. Also, the table has aggregates for each row and each column. |

# Product Backlog Stories

|  |  |  |
| --- | --- | --- |
| Backlog # | User Story # | Backlog Item Description |
| 1. | 1. | Query database with search term/s according to the Search Form data filled out |
| 2. | 1. | Parse customer(s) account matching search query with photo, name, age and phone |
| 3. | 1. | Show relevant customer profile on customer selected |
| 4. | 2. | Display results from query |
| 5. | 2. | Query database for specific customer information stored in account |
| 6. | 2. | Parse query results into Detailed Form with customer’s photo, full name, date of birth, allergies, emergency contact information, and swimming ability. |
| 7. | 3. | Allow “Check-in” functionality from Search Form and Detailed Form |
| 8. | 3. | Write check-in event into system logs |
| 9. | 3. | Update customer in current swimmers list |
| 10. | 4. | “Add new” customer functionality depending on user privilege level using Detailed Form |
| 11. | 4. | Capture store customer information and store into database |
| 12. | 4. | Validate form completion and data |
| 13. | 5. | Enable Detailed Form to be editable depending on user privilege level |
| 14. | 5. | Capture new data from user |
| 15. | 5. | Update customer in database |
| 16. | 5. | Validate form completion and data |
| 17. | 6. | Query database for current swimmers form, which displays customers who are currently checked-in |
| 18. | 6. | Parse query results into Status Form |
| 19. | 6. | Give “refresh” capabilities for up-to-date results |
| 20. | 7. | Add “Check-out” functionality to Status Form, Search Form, or Detailed Form. |
| 21. | 7. | Write check-out event into system logs, tracking relevant information such as check-in time, check-out time, and number of swimmers in group |
| 22. | 7. | Delete customer from current swimmers list |
| 23. | 8./9./10. | Reports screen allows all users to build a Visits query with information such as customer or date and time range, or Attendance query with date range and selected statistic |
| 24. | 8./9./10. | Validate query terms with error catching and display appropriate error messages; execute queries into appropriate database |
| 25. | 8./9. | Visits option allows users to build a query specifying a time range (start and end date/time) and target population (specific customer or all). |
| 26. | 8./9. | Parse query results into a Visits Report to view a given population’s number of visits within the given time range |
| 27. | 10. | Attendance option allows all users to build a query specifying a time range (start and end date/time), a selection of either minimum, average, maximum, or total (referring to current swimmers) |
| 28. | 10. | Parse query results into form Attendance Report, displaying days of week as column names, hours of day as rows, and totals. |

# Sprints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint #1: from 2 OCT 2018 to 14 OCT 2018 | | | | | |
| Backlog # | **Task Description** | **Story Point Estimate** | **Story Point Actual** | **Status\*** | **Comments** |
| \* | Create sign-up form for username, user info, password, and access level | 2 |  |  |  |
| \* | Capture and store data from form in database | 1 |  |  |  |
| \* | Provide error checking on form submission: redundant username, invalid passwords, etc. | 1 |  |  |  |
| \* | Develop log-in panel to enter username and password, sign-up, or delete user | 1 |  |  |  |
| \* | Develop navigational panel to direct users to user case options | 1 |  |  |  |
| \* | Provide error checking on username/password entry; display error messages | 1 |  |  |  |
| \* | Develop delete user option that asks for confirmation with re-input of password | 1 |  |  |  |
| \* | Create necessary prototype files to simulate database (JSON files and customer images) | 1 |  |  |  |
| \* | Test username access and functionality | 1 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint #2: from 15 OCT 2018 to 28 OCT 2018 | | | | | |
| Backlog # | **Task Description** | **Story Point Estimate** | **Story Point Actual** | **Status\*** | **Comments** |
| 10 | Develop input form logic, values, and design | 4 |  |  |  |
| 11 | Develop read and write functionality to JSON file | 1 |  |  |  |
| 12 | Add error checking to input form and try/catch blocks to write/read processing | 1 |  |  |  |
| 13 | Develop an option to recall the input form for existing customers | 1 |  |  |  |
| 14 | Form displays pre-existing data that can be deleted and re-entered | 1 |  |  |  |
| 15 | Override existing data with new data once update is submitted | 1 |  |  |  |
| 16 | Produce “success” message or suggest unfilled inputs or error diagnosis | 1 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint #3: from 29 OCT 2018 to 4 NOV 2018 | | | | | |
| Backlog # | **Task Description** | **Story Point Estimate** | **Story Point Actual** | **Status\*** | **Comments** |
| 1 | Develop search criteria and algorithms | 1 |  |  |  |
| 2 | Build logic that filters customers from database and produces results according to search | 1 |  |  |  |
| 4 | Build display panel capable of showing data from search | 1 |  |  |  |
| 3 | Create functionality to click on customer in search result and open profile panel | 1 |  |  |  |
| 6 | Develop profile panel to organize and display customer info and picture | 2 |  |  |  |
| 5 | Feed in data on selected customer to profile panel | 1 |  |  |  |
| 7 | Add check-in status variable to customers and check-in buttons | 1 |  |  |  |
| 8 | Capture check-in times through running log | 1 |  |  |  |
| 9 | Create list of current swimmers with logic to add/remove given check-in status | 1 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint #4: from 5 NOV 2018 to 18 NOV 2018 | | | | | |
| Backlog # | **Task Description** | **Story Point Estimate** | **Story Point Actual** | **Status\*** | **Comments** |
| 17 | Build swimmers’ panel to display thumbnails of checked-in customers | 2 |  |  |  |
| 18 | Use list of active swimmers to populate swimmers’ panel with data | 1 |  |  |  |
| 19 | Add ‘refresh’ functionality and add button to reload panel | 1 |  |  |  |
| 20 | Add ‘check-out’ functionality and add button to each thumbnail | 1 |  |  |  |
| 21 | Update check-in log to incorporate check-out events | 1 |  |  |  |
| 23 | Test that check-out status is reflected in the removal of thumbnail from swimmers’ panel | 1 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint #5: from 19 NOV 2018 to 9 DEC 2018 | | | | | |
| Backlog # | **Task Description** | **Story Point Estimate** | **Story Point Actual** | **Status\*** | **Comments** |
| 23 | Develop reports query panel to display different query options: Visits and Attendance | 3 |  |  |  |
| 24 | Implement error catching to define valid date ranges and customers | 1 |  |  |  |
| 25 | Build logic to accept date range criteria and target population; return log of all relevant visits within range | 1 |  |  |  |
| 26 | Develop resulting visits panel to display visits of population from date range | 1 |  |  |  |
| 27 | Build logic to calculate totals, averages, maximums, and minimums from visits by hour and day | 1 |  |  |  |
| 27 | Build logic to accept date range criteria and return attendance statistics as specified | 1 |  |  |  |
| 28 | Create attendance report table to graphically represent daily and hourly statistics | 2 |  |  |  |