* **Introduction** 
  1. Electronic Form-1
  2. Gavin Delphia, Austin Hurley
  3. Superintendents, Squadron Commanders
* **Requirements**
  1. The software shall allow the superintendent to register with a USAFA email.
  2. The system shall email the superintendent a validation code upon legal registration.
  3. The superintendent shall enter the validation code ensuring email ownership.
  4. The software shall allow the superintendent to log in.
  5. The software shall allow the squadron commander to register with a USAFA email.
  6. The system shall email the squadron commander a validation code upon legal registration.
  7. The squadron commander shall enter the validation code ensuring email ownership.
  8. The software shall allow the squadron commander to log in.
  9. The squadron commander shall only have read-only privileges on the software.
  10. The superintendent shall be able to create a list of emails that can be on a form-1.
  11. The software shall parse the emails into cadet table database rows.
  12. The software form-1 shall display the cadets in the database.
  13. The software form-1 shall organize the cadets in the form-1 by class year.
  14. The superintendent shall be able to add names to the cadet table.
  15. The superintendent shall be able to remove names to the cadet table.
  16. The superintendent shall be able to create a new instance of a form-1 with event information.
  17. The superintendent may be able to modify a form-1.
  18. The superintendent may be able to delete a form-1.
  19. The superintendent shall be able to click on the names of cadets not at the event.
  20. The superintendent shall be able to remove the absent status of a cadet.
  21. The superintendent shall be able to request people marked absent be sent an email.
  22. The software shall email cadets that were marked absent from a mandatory event.
  23. The superintendent shall submit a form-1 for dissemination.
  24. The software shall email the chain of command the results of accountability.
  25. The superintendent shall be able to log out of the system.
  26. The squadron commander shall be able to log out of the system.
  27. The system shall terminate a connection upon log out.

**Use Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario:** | *Step* | *Action* | *Key Information Elements* | *Data Source(s)* |
| 1. | User wants to use the easySupt software | easySupt web site URL | Externally provided |
| 2 | User is immediately able to read about the easySupt software. | Provided description of easySupt software. | Software designers |
| 3. | User tries to register | Email address, password, squadron # | The user |
| 4. | Upon user first login the system attempts to verify their email. | Verification code, email, password | User provided, System provided |
| 5. | User populates squadron cadet emails to the software. | Cadet emails separated by semicolon. | Outlook GAL |
| 6. | User wants to edit, delete or add a cadet to the roster. | Cadet email or desired change. | The user |
|  | 7. | The user wants to create a new form-1 | Event name and date | The user |
|  | 8. | The user wants to view a past form-1 | Clicking on the appropriate event link | The user and the software |
|  | 9. | The user wants to track who was missing / present. | Accountability and the cadet name | The user |
|  | 10. | The user requests excusals form absent cadets. | The absent cadets and their emails. | The user and the software. |
|  | 11. | The user wants to notify chain of command the accountability | Email address of squadron commander, accountability results | The software |
|  | 12. | The user wants to logoff the server. | The logoff button | The software |
|  | 13. | The user is a read-only user and wants to make changes but is denied. | The software user controls | The software |

**Development Plan**

We plan on doing a combination of pair programming for the more difficult tasks and divide and conquer for the simpler tasks. For the divide and conquer we will be using Dropbox to ensure we always have the most up to date changes.

**Spiral 1**

* UI / CSS completed
* Client-side javascript validation
* Can establish a connection to the database
* User login/register capability
* Code boilerplate

**Spiral 2**

* Test results all pass for the functional requirements
* Completely working application
* Updated design documents to reflect any changes
* Personal reflection
* Peer evaluation

**Test Plan**

* The test environment will be the localhost running on one of our computers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement #** | **Procedure** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1 | The superintendent inputs their email, password and squadron and clicks register on the register page. | Server will send a validation code to the superintendent email address and return to index.html. It will also save this information to the database. | Superintendent is registered with the system | Pass |
| 2 | A superintendent legally registers. | The superintendent actually receives the code in an email. | Change of design of system removed this requirement from implementation | N/A |
| 3 | The superintendent enters their email, password and validation code. Then clicks login. | The server accepts the validation code and annotates the superintendent as verified. | Change of design of system removed this requirement from implementation | N/A |
| 4 | The superintendent attempts to login after the correct email and password are provided a login is clicked. | The server verifies the email and password combination and takes the squadron commander to the main page. | Superintendent is logged in | Pass |
| 5 | The squadron commander inputs their email, password and squadron and clicks register on the register page. | Server will send a validation code to the squadron commander email address and return to index.html. It will also save this information to the database. | Squadron commander is registered | Pass |
| 6 | A squadron commander legally registers. | The squadron commander actually receives the code in an email. | Change of design of system removed this requirement from implementation | N/A |
| 7 | The squadron commander enters their email, password and validation code. Then clicks login. | The server accepts the validation code and annotates the squadron commander as verified. | Change of design of system removed this requirement from implementation | N/A |
| 8 | The squadron commander attempts to login after the correct email and password are provided a login is clicked. | The server verifies the email and password combination and takes the squadron commander to the main page. | Squadron commander is logged in | Pass |
| 9 | The squadron commander is directed to the form-1 page | The form-1 page does not display the administrative buttons | After logging in, any users are directed to the dash page | Pass |
| 10 | The superintendent correctly populates the email field and presses submit. | If the emails are separated by semicolons, the software populates the cadet table in the database for the email column. | Emails separated by semicolons entered into field | Pass |
| 11 | Upon user clicking submit with valid emails. | The software correctly parses out cadet first name, last name, class year and those values are put into columns in the cadet table of the database. | Software correctly parses emails into cadet table | Pass |
| 12 | When a user clicks on a past or new form-1. | The software correctly displays the cadet information in the form-1. | Software displays correct information in form 1 | Pass |
| 13 | When a user clicks on a past or new form-1. | The software correctly organizes cadets into the correct class year columns. | The software correctly organizes cadets into the correct class year columns. | Pass |
| 14 | The superintendent inputs another valid email into the email field on the roster page. | The software correctly parses and adds the first name, last name, class year and email to the cadet table columns in the database. | The software correctly parses and adds the first name, last name, class year and email to the cadet table columns in the database. | Pass |
| 15 | The superintendent clicks on a name and selects modify. | The software prompts for the name change and correctly updates the cadet table in the database. | Change of design of system removed this requirement from implementation | N/A |
| 16 | The superintendent clicks the new form-1 button and inputs the event name and date then clicks the request excusal button. | The software adds the form-1 to the event table and builds the form table entry. | The software adds the form-1 to the event table and builds the form table entry. | Pass |
| 17 | The superintendent modifies data on the form-1 and clicks the save button. | The software makes the change and reflects them in the database but does not email anyone. | The software makes the change and reflects them in the database but does not email anyone. | Pass |
| 18 | The superintendent presses the delete button next to a past form-1. | The software confirms the request for deletion and upon success deletes the form-1 and associated database information and upon canceling does nothing. | The software confirms the request for deletion and prevents non admins from deleting form 1s. | Pass |
| 19 | The superintendent clicks on a specific cadet’s name. | The software changes the status from present if it was not present and vice versa. | The software changes the status from present if it was not present and vice versa. | Pass |
| 20 | The superintendent clicks on an absent marked cadet. | The software changes the cadet’s status from absent to present. | The software changes the cadet’s status from absent to present. | Pass |
| 21 | The superintendent clicks on the request excusal button. | The software emails all absent marked cadets a request for an excusal. | The software creates an email to all absent marked cadets a request for an excusal. | Pass |
| 22 | The software sends a list of emails to the mail server with a message. | The mail server emails all of the provided email addresses with the desired message. | Change of design of system removed this requirement from implementation | N/A |
| 23 | The superintendent clicks the submit accountability button. | The software emails the accountability report to the chain of command. | Change of design of system removed this requirement from implementation | N/A |
| 24 | The software sends a list of the emails to the mail server with the accountability report. | The mail server emails all of the provided email addresses with the desired message. | Change of design of system removed this requirement from implementation | N/A |
| 25 | The superintendent presses the log out button. | The software logs off the superintendent and returns to index.html. | The software logs off the superintendent and returns to index.html. | Pass |
| 26 | The squadron commander presses the log out button. | The software logs off the squadron commander and returns to index.html. | The software logs off the squadron commander and returns to index.html. | Pass |
| 27 | A user clicks the log off button. | The system properly terminates the login session and returns to index.html. | The system properly terminates the login session and returns to index.html. | Pass |

BUGS: We are not aware of any ☺

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_

CS 364 Final Project Specification  
Cut Sheet

## Turn-in Requirements

\_\_\_\_\_ Appropriate Documentation provided (-3 if not)

\_\_\_\_\_ Electronic file submission requirements followed (-3 if not)

## Assignment Requirements

|  |  |
| --- | --- |
| **Points** |  |
|  | (5) Introduction |
|  | (25) Requirements (enumerated, elaborated (DFD or Use Case(s)) |
|  | (25) Design (ERD, user interface, component structure chart, interface definition) |
|  | (10) Test Plans (test environment, test procedures) |
|  | (10) Development Plans (team work breakdown, spiral definitions) |
|  | (0-5) BONUS points for exceptional work |
|  | **TOTAL** |

Documentation: None