## Scrumwise group name: SD3GroupB

## Git repository link:

## [*https://github.com/mcgeough/Jet2Go-Y3-Software-Project*](https://github.com/mcgeough/Jet2Go-Y3-Software-Project)

## Functionality

**Expected aspects: Demonstrate minimum of three core pieces of functionality developed for your system – these must include login, register and a view/display of some information from your database.**

### Completed functions:

*<Insert list of fully functional elements>*

### Partial/Incomplete functions:

*<Insert list of incomplete elements of functionality>*

### Broken functions:

*<Insert list of known bugs>*

## Database

**Expected aspects: Minimum 5 tables, well designed to support your project as a whole. Consideration given to the relationships or composition of the different entities. Consideration given to use of unique fields, choice of keys and appropriate selection of field types.**

### ER Diagram:

*<Insert diagram of your database design to date>*

## User Experience (UX)

**Expected aspects: Easy to use, looks appropriate, well-designed, consistent across all areas of the site.**

### Outline of special considerations:

<Insert list of any special consideration/features included to improve design>

## Security

**Expected aspects: Document authentication process and control of access to restricted areas in your site (e.g. JWT to secure all requests). Ensure no clear passwords can be accessed if you store them in your DB. Where appropriate use security services available in your platform (AWS, Spring etc).**

### Outline of implemented features:

*<Insert list of implemented security features to date>*

## Validation

**Expected aspects: Client-side AND server-side validation of information (is required information present, is it a valid structure, password rules etc).**

### Outline of implemented features:

*<Insert list of examples of different types of validation included>*

## Testing

**Expected aspects: JUnit testing of DAO classes (excluding DTOs, interfaces, DAO connection/superclass). Mocking is preferable but integrated components (i.e. methods accessing the database instead of calling on a mock object) are acceptable. Consider system testing approach (e.g. Postman for REST interface).**

### Outline of implemented tests:

*<Insert list outlining extent of testing implemented>*

Note: There must be SOME tests coded.

## Project Management

**Expected aspects: Creation of release plan, multiple iterations with tasks created and tracked through their stages of progress, team member contribution tracking etc.**

This should be documented on Scrumwise in line with Agile principles.