CSCC01

Project Deliverable #3

Java Bean (Team28)

Brian Chim

Vladlen Lyudogovs'ky

Abithan Kumarasamy

Matthew Varga

Maggie Zhou

Table of Content

Sprint 01

Sprint 02

*(no update to Personas)

Sprint01

Task breakdown

User Story 1: As Charlie (an Admin), I would like to create an account, so various users can use the application.

Task 1 – project setup

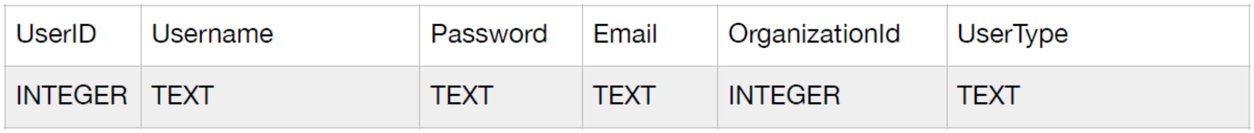
- story points: 1; dependency: n/a
- setup boilerplate of project including maven dependencies for sqlite and javafx
- create project folder structure for main/test

Task 2 – create database schema for selected stream

- story points: 1; dependency: n/a
- create database schema for EmployeeServiceStream using sqlite viewer and this will be the only stream supported for User Story 1
- add an extra column in the User Table, and it should indicate if a user has access to it or not
- EmployeeServiceStream schema can be extracted from iCare template from Piazza

Task 3 – add user table schema in database

- story points: 1; dependency: Task 2
- users will have the following fields:
  - Username: name of user
  - Password: password used for authentication
  - Email
  - Organization ID: the ID to distinguish between different agencies
  - UserType: one of the following - ADMIN, TEQ STAFF, AGENCY



UserID	Username	Password	Email	OrganizationId	UserType
INTEGER	TEXT	TEXT	TEXT	INTEGER	TEXT

Task 4 – implement user classes

- story points: 2; dependency: Task 1
- create different user classes to handle the different types of users of the system
- each User should have the following attributes: userID, username, password, orgID, orgType (AGENCY/ADMIN/STAFF), list of given permissions, list of available service streams
  - AGENCY users should have PERMISSION to UPLOAD files
  - STAFF users should have PERMISSION to GENERATE SUMMARY AND TREND reports

    - ADMIN users should have PERMISSION to GENERATE SUMMARY AND TREND reports and CHANGE ACCESS LEVELS
- add getters and setters for user classes

Task 5 – implement DBHandler with user insert/get methods

- story points: 5; dependency: Task 4
- create a DBHandler that has a insertUser method and getUser method
- insertUser (HashMap userDetails, ArrayList streams) where the hash map should have all the database schema columns as strings mapped to their respective values
- getUser(String username, String password) which should authenticate the user and return a hashmap with all the info from the databse regarding the user's details
- connect method to create connection to database

Task 6 – create registration form

- story points: 5; dependency: Task 1
- setup ui folder structure and main entry of the application user interface
- create registration form with proper field visibilities for different types of users:
    - ADMIN user only need to input username, password and user type, TEQ STAFF user needs to select organization and AGENCY users will need to select their agencies and service streams

Task 7 – connect registration form with database for user insertion

- story points: 3; dependency: Task 5, Task 6
- collect form data from the registration form and perform completion checking -> all the displayed fields for different users are mandatory
- make call to DBHandler.insertUser method with formatted inputs
- generate pop-up windows to notify users of the signup status (successful/failed)

User Story 2: As Alice (an Agency Representative), I would like to upload an iCare excel file, so that and have the user's information stored securely so that I do not worry about losing my local copy

Task 8 – implement file parser

- story points: 5; dependency: n/a
- use apache.poi library for parsing excel files in the format of the iCare template provided by client
- method returns a HashMap of <Column<String>, Value<String>>

Task 9 - Create Login page + authenticate user

- story points: 8; dependency: Task 5
- login page should include the following:
    - Text INPUT field for the USERNAME
    - Text INPUT field for the PASSWORD (must be masked)
    - Button to login which VALIDATES whether the USERNAME and PASSWORD match the user in the DB
    - Button to register which opens the REGISTER page in a NEW OVERLAY
    - Appropriate ERROR MESSAGE if USERNAME and PASSWORD do NOT MATCH. User should be able to RETRY
- if user successfully logs in, create new stage and pass in the User created (new stage to be done)

Task 10 – add service stream schemas

- story points: 3; dependency: Task 2
- add the remaining service stream schemas to the database as tables
- the service streams are listed as follows:
    - needs assessment & referrals
    - community connections
    - info & orientation
    - language training client enrol
    - language training course setup
    - language training client exit

Task 11 – create select file UI

- story points: 5; dependency: Task 4
- create the select file UI
- should be a button that opens up window allowing you to choose a file for upload
- returns a filepath for selected file

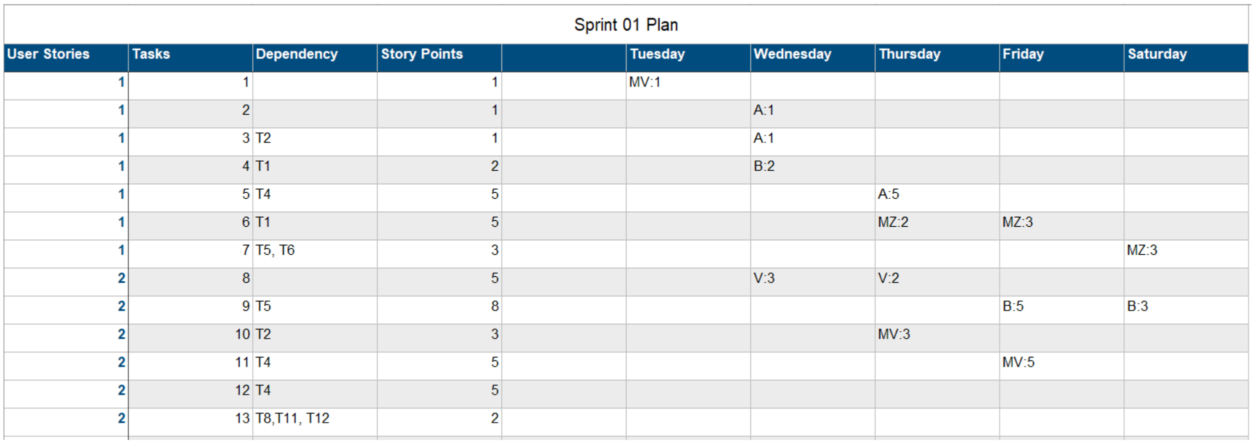
Task 12 - create dbhandler for upload method

- story points: 3; dependency: n/a
- Create DBhandler method to insert data into EmpolymentServiceStreamTable
- Checking is assumed to have been completed before function call, so it pushes the data as is
- Data should come in a hashmap <String, String>, keys should match column name

Task 13 - insert parsed file info into database

- story point: 2; dependency: Task 8, Task 11, Task 12
- make DBHandler call to insert parsed file info into database
- get the insertion status back (true/false)

Sprint 01 Planning



Sprint 01 Plan

User Stories	Tasks	Dependency	Story Points		Tuesday	Wednesday	Thursday	Friday	Saturday
1	1		1		MV:1				
1	2		1			A:1			
1	3	T2	1			A:1			
1	4	T1	2			B:2			
1	5	T4	5				A:5		
1	6	T1	5				MZ:2	MZ:3	
1	7	T5, T6	3						MZ:3
2	8		5			V:3	V:2		
2	9	T5	8					B:5	B:3
2	10	T2	3				MV:3		
2	11	T4	5					MV:5	
2	12	T4	5						
2	13	T8,T11, T12	2						

*MV = Matthew Varga

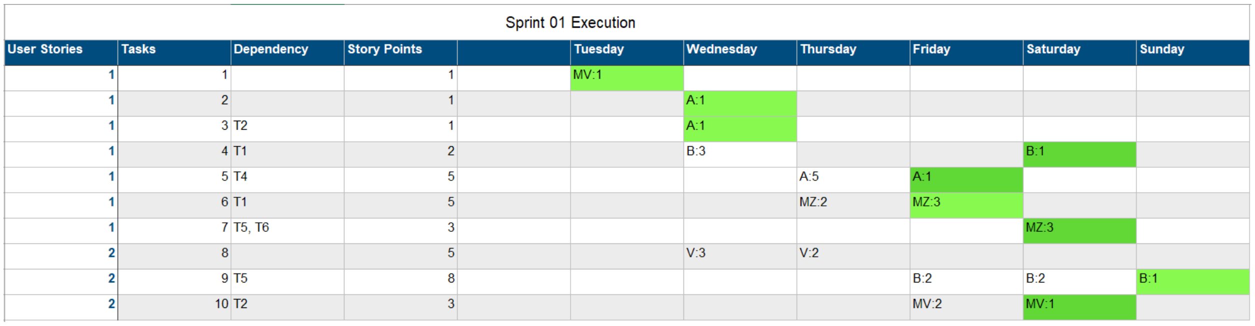
A = Abithan Kumarasamy

B = Brian Chim

MZ = Maggie Zhou

V = Vladlen Lyudogovs’ky

Sprint 01 Report – Execution



Sprint 01 Execution

User Stories	Tasks	Dependency	Story Points		Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	1		1		MV:1					
1	2		1			A:1				
1	3	T2	1			A:1				
1	4	T1	2			B:3			B:1	
1	5	T4	5				A:5	A:1		
1	6	T1	5				MZ:2	MZ:3		
1	7	T5, T6	3						MZ:3	
2	8		5			V:3	V:2			
2	9	T5	8					B:2	B:2	B:1
2	10	T2	3					MV:2	MV:1	

*green cell marks the completion of a task

Progress Reports:

Tuesday (10/16): Meet expectation

Wednesday (10/17): Meet expectation

Thursday (10/18):

- Task4: the original scope for Task 4 was completed as expected, but the design decision was reconsidered later in the week which resulted in extra working hours
- Task 5: extra hour required
- Task 8: encountered technical challenge using parsing library
- Task 10: needed to move working hour due to personal issue

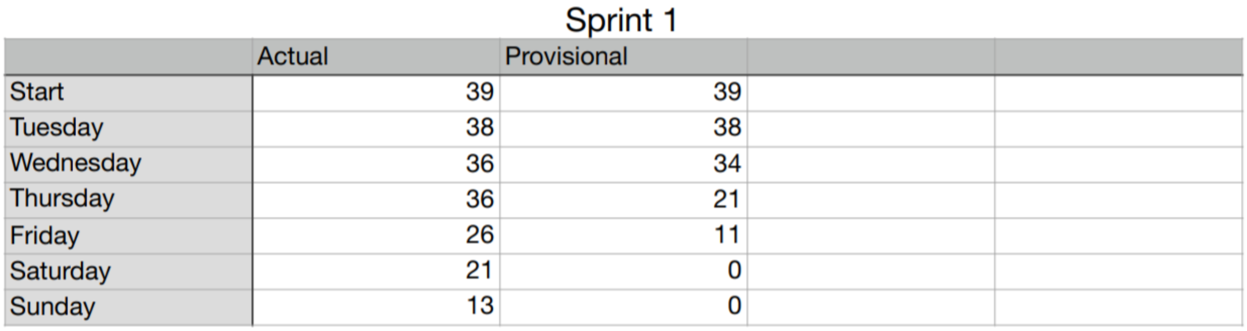
Friday (10/19):

- Meet expectation except that Task 11 was pushed back to the next sprint as result of the schedule change from the previous day

Saturday (10/20):

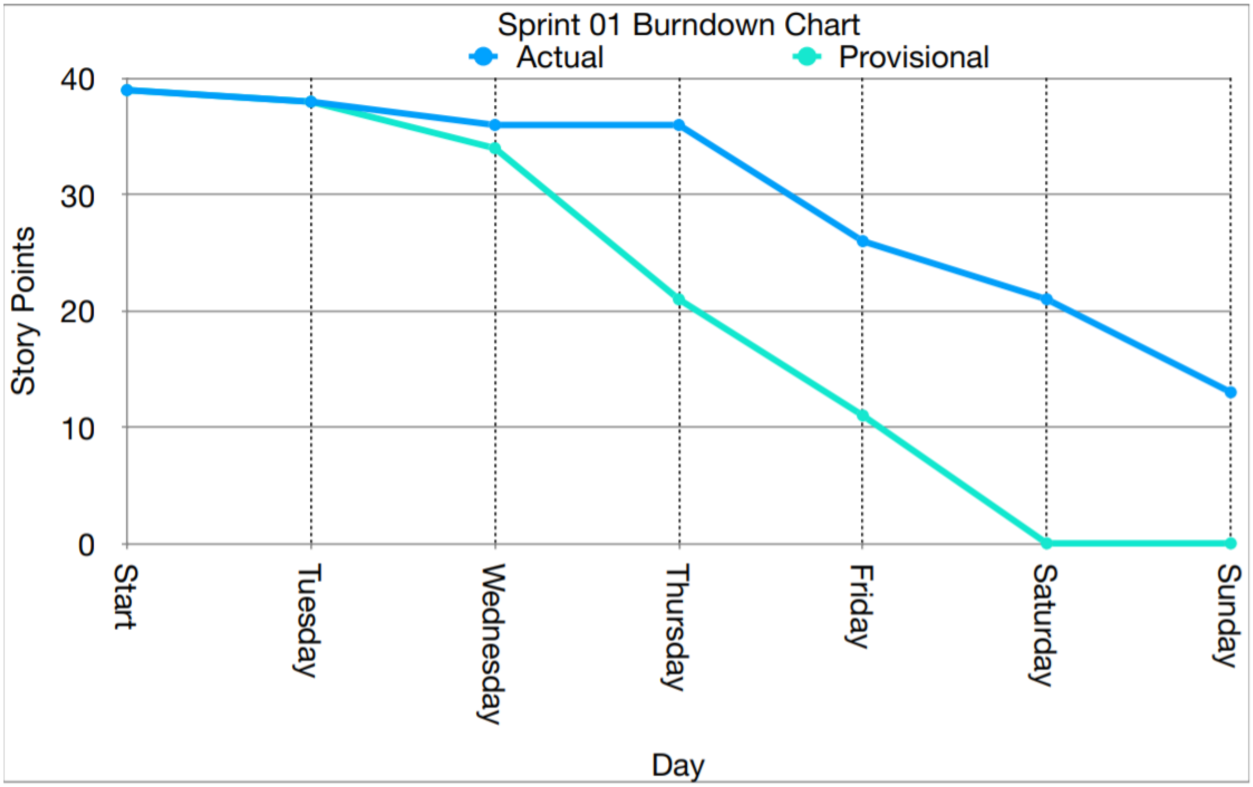
- Completion for Task 9 was pushed back due to the rework for Task 4 as mentioned above

Burndown Chart



Sprint 1

	Actual	Provisional		
Start	39	39		
Tuesday	38	38		
Wednesday	36	34		
Thursday	36	21		
Friday	26	11		
Saturday	21	0		
Sunday	13	0		



Sprint02

Task breakdown

User Story 2 (continue)

Task 11 – create select file UI

- story points: 5; dependency: n/a
- create a ui tab for uploading a file
- add two buttons, one for uploading a single file, one for uploading multiple files
- open file selector when clicking buttons
- add a dropdown selector for the service stream you are uploading to

Task 12 - create dbhandler for upload method

- story points: 3; dependency: n/a
- Create DBhandler method to insert data into EmpolymentServiceStreamTable
- Checking is assumed to have been completed before function call, so it pushes the data as is
- Data should come in a hashmap <String, String>, keys should match column name

Task 13 - insert parsed file info into database

- story point: 2; dependency: Task 8, Task 11, Task 12
- make DBHandler call to insert parsed file info into database
- get the insertion status back (true/false)

Task 14 – create main application page

- story points: 3; dependency: n/a
- create a tab pane to host all the tabs for specific user permissions
- create a home tab with some generic content
- ensure tab pane exists for all types of users

User Story 3: As Ellie (an Agency Analyst), I would like for the application to let me know when it was unable to merge a field, so that i can fix the issue and reupload

Task 15 – upload button UI + error table

- story points: 5; dependency: Task 13
- create a ui tab for updating a file

- add a buttons for uploading a file
- open file selector when clicking the button
- add an area below button to display any errors like row/col that had merge conflicts

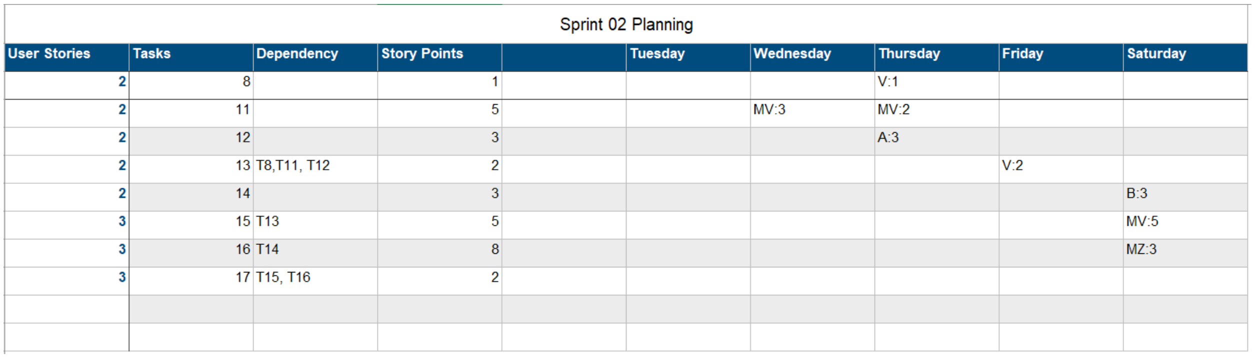
Task 16 – implement conflict checking before file info insertion

- story points: 8; dependency: Task 14
- create SafeUploader class inside util package which checks if parsed file info has conflict info in the database
- loop through database to check if info for a particular client has already existed, if so, inform the user about the conflicting clientID

Task 17 – connect upload logic to UI

- story points: 2; dependency: Task 15, Task16
- add upload button click logic to UI
- test upload workflow with an icare template

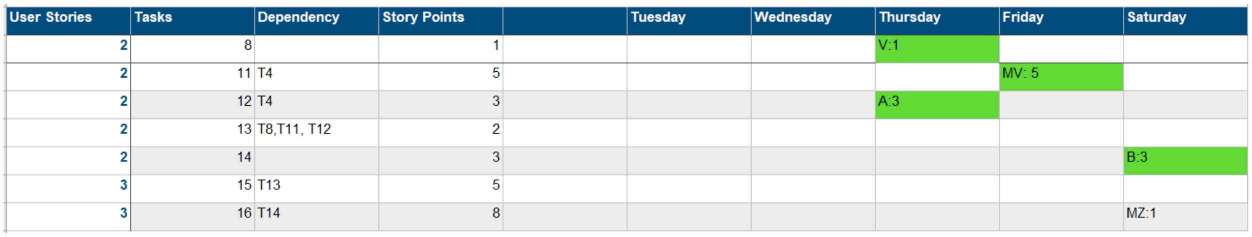
Sprint 02 Planning



Sprint 02 Planning

User Stories	Tasks	Dependency	Story Points		Tuesday	Wednesday	Thursday	Friday	Saturday
2	8		1				V:1		
2	11		5			MV:3	MV:2		
2	12		3				A:3		
2	13	T8,T11, T12	2					V:2	
2	14		3						B:3
3	15	T13	5						MV:5
3	16	T14	8						MZ:3
3	17	T15, T16	2						

Sprint 02 Report – Execution



User Stories	Tasks	Dependency	Story Points		Tuesday	Wednesday	Thursday	Friday	Saturday
2	8		1				🟩 V:1		
2	11	T4	5					🟩 MV: 5	
2	12	T4	3				🟩 A:3		
2	13	T8,T11, T12	2						
2	14		3						🟩 B:3
3	15	T13	5						
3	16	T14	8						MZ:1

*green cell marks the completion of a task

Progress Reports:

Wednesday (10/24):

- Task 11: pushed back because of personal issue

Thursday (10/25):

- Meet expectation except that Task 11 was pushed back

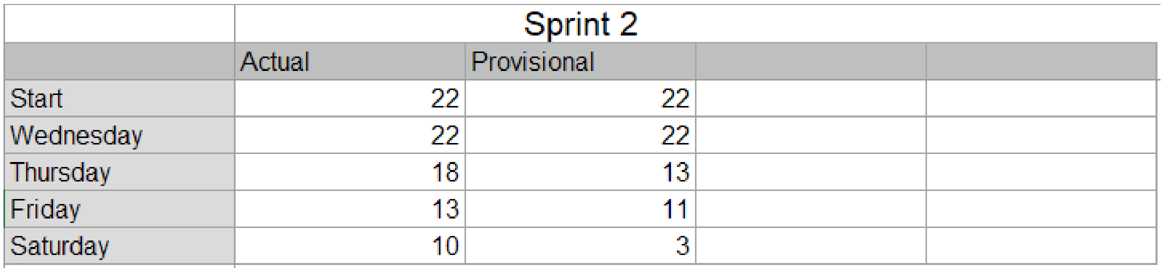
Friday (10/26):

- Task 13: not clear about the implementation requirements, decided to push back

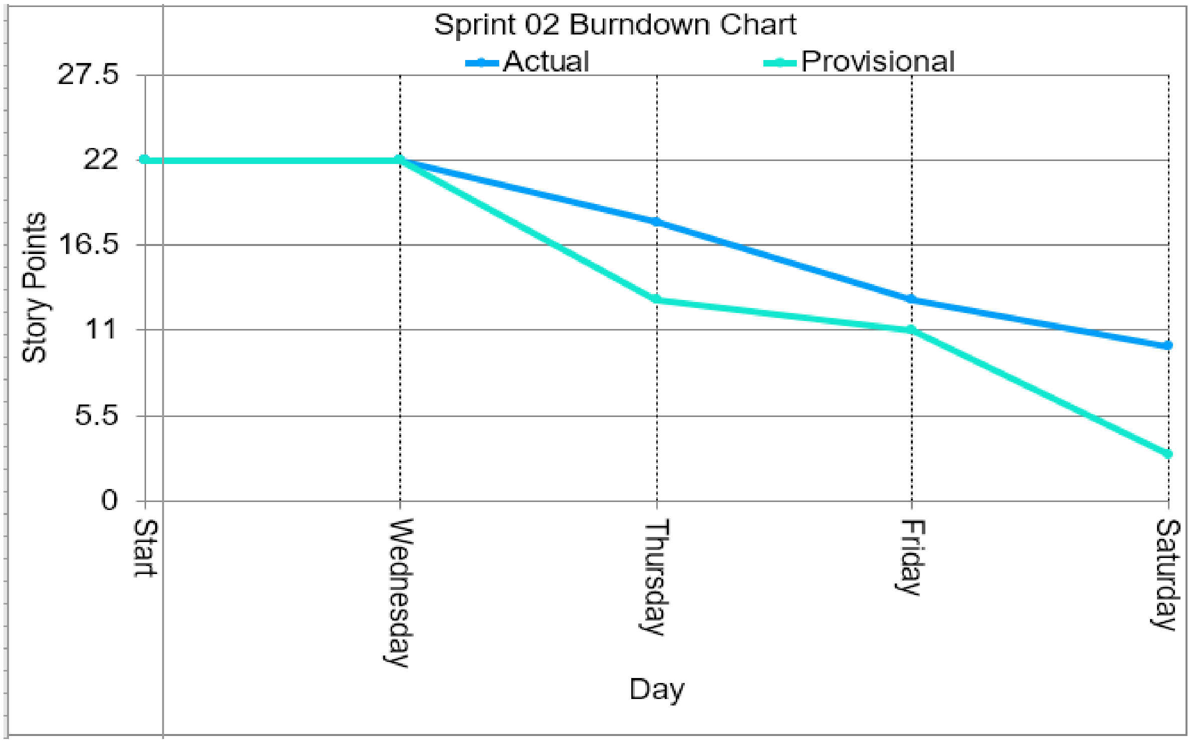
Saturday (10/27):

- Task 15: pushed back because of other course work
- Task 16: not working enough hour because of other course work

Burndown Chart



	Sprint 2			
	Actual	Provisional		
Start	22	22		
Wednesday	22	22		
Thursday	18	13		
Friday	13	11		
Saturday	10	3		



User Stories (Version 1)

1. 1. As Charlie (an Admin), I would like to create an account, so various users can use the application.

2. As Alice (an Agency Representative), I would like to upload an iCare excel file, so that and have the user's information stored securely so that I do not worry about losing my local copy

3. As Ellie (an Agency Analyst), I would like for the application to let me know when it was unable to merge a field, so that i can fix the issue and reupload.

4. As David (CCS organization manager), I would like to generate reports (summary and trend), so that I can use the report to improve future newcomer service.

5. As Alice (an Agency Representative), I want to be able to remove a certain user from the application, as they may no longer require any newcomer services or their circumstances have changed.

6. As Ellie (an Agency Analyst), I would like for the application to automatically merge postal codes and phone numbers if the only difference is their format, so that I don't need to worry which format the first uploaded document was in.

7. As Alice (an Agency Representative), I want to be able to upload multiple Excel spreadsheets to the system at a time, so that I can save time when uploading information.

8. As Bob (an Agency Supervisor), I would like to generate my employees history, so I can monitor what changes were made and which employee made them in order to identify issues quickly if they arise.