**Iteration 1**

**Group Name**

**CSCI 3130 Group 4**

**Group Members**

**Nathanael Bowley**

**John Corsten**

**Nathan Horne**

**Ted Graveson**

**Ding Ding**

**Tianhao Jia**

**Hongzheng Ding**

**Saher Anwar Ziauddin**

**Date: Febuary 5th, 2022**

**Mention your GitLab repo:** https://git.cs.dal.ca/corsten/csci-3130-group-4-project

**Iteration Plan**

1. **User Stories & Acceptance Criteria**

**(Reformate into correct format for user stories)**

* 1st -> log in functionality as a user if i want to log into app i should be able to if i want to use it email and password 🡪
  + As a User, I want to be able to enter an email and password so that I can log in.
* 2nd -> if new user then register (separate for Employee and Employer) info name, email, address,

phone number, password firebase to store and retrieve info

* + As a New User, I want to be able to register and provide my name, email address, phone number, password, and choose Employee or Employee so that my information will be stored for later login.
* 3rd -> if want to log in as employee - directed to a diff page and like wise a diff page for employees
  + As a User, I want to be able to be directed to a different landing page depending on if I’m an employee or employer, so that I see the correct UI activity.
* 4th -> log out if loged in they should be able to log out if logged in
  + As a User, I want to be able to log out if I have logged in so that I can protect my information.
* 5th -> should be able to stay logged in
  + As a User, I want to be able to stay logged in if already logged in, so that if I switch applications I don’t get logged out and have to re-login.
* 6th -> when the user is logged in the user should be directed to landing page for now main page is a map that you will be adding functionality later
  + As a User, I want to be directed to a landing page so that I can see details of the app functionality later.

**Provide your git project dashboard link.**

**2. Tasks Assignment: Acceptance test and Engineering task**

**[Necessary Field --> labels,task description, member assigned, start date, end date(expected), status [pending,completed,in progress] ]**

* **Could be downloaded from gitlab [refer tutorial]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Corresponding UserStory and Acceptance Criteria** | **Task(s)** | **Estimate** | **Member Assigned** | **Start** | **Expected** | **Status** |
| AT:1:1 | Implement Initial Login Functionality | 4hrs | Member 1 | 8th June, 2019 | 12th June, 2019 | Completed |
| AT:1:1 | Write Espresso Tests | 30 minutes | Member 1 | 8th June | 8th June | Completed |
| AT:1:1 | Refactor Espresso Tests | 30 minutes | Member 2 | 12th June | 12th June | Completed |
| AT:1:2 | Create Landing page from Login | 30 mins | Member 2 | 12th June | 12th June | Completed |
| AT:1:3 | Adding a User class | 1hr | Member 2 | 12th June | 17th June | Completed |
| AT:1:3 | Refactor Login Code | 1hr | Member 2 | 17th Jun | 17th June | Completed |
| AT:3:1 | Write Espresso Tests, ran the test, but failed. | 1hr | Member 3 | June 10th | June 11th | Completed |
| AT:3:1 | Implement checkViewFieldsAndUpdateMedicationObject() method to check the validity of day and time. | 1 hr | Member 3 | June 10th | June 11th | Completed |
| AT:3:1 | Update the logic to make sure changes are consistent | 30 minutes | Member 3 | June 10th | June 11th | Completed |
| AT:3:1 | Run the test, and pass. | 30 minutes | Member 3 | June 10th | June 11th | Completed |
| AT:3:1 | Run acceptance tests, and pass. | 30 minutes | Member 3 | June 10th | June 11th | Completed |
| AT:3:1 | Code documentation, to have the complete Java doc, meaningful variable and method names, inline comments where necessary. | 30 minutes | Member 3 | June 10th | June 11th | Completed |
| AT:5:1 | Write Unit Tests For Medication Class | 1hr | Member 3 | June 7th | June 9th | Completed |
| AT:5:1 | Implement Medication Class | 30 minutes | Member 3 | June 7th | June 9th | Completed |
| AT:5:1 | Add Button Functionality to Access Medication List Page | 30 minutes | Member 3 | June 8th | June 9th | Completed |
| AT:5:1 | Build UI For Medication List | 1hr | Member 3 | June 8th | June 9th | Completed |
| AT:5:1 | Connect UI to Database | 1hr | Member 3 | June 8th | June 9th | Completed |
| AT:5:1 | Build UI to Add Medication | 30 minutes | Member 3 | June 8th | June 9th | Completed |
| AT:5:1 | Take User Input and Send to Database | 1hr | Member 3 | June 9th | June 11th | Completed |
| AT:5:1 | Write Espresso Tests | 1hr | Member 3 | June 9th | June 11th | Completed |
| AT:5:1 | Refactor to separate business/UI logic | 1hr | Member 3 | June 9th | June 11th | Completed |
| AT:6:1 | Write Espresso tests | 1 hr | Member 4 | June 17th | June 17th | Completed |
| AT:6:1 | Run tests, but tests failed, figure why, and make small changes | 2.5 hr | Member 4 | June 17th | June 17th | Completed |
| AT:6:1 | Update the logic to remove medication from the list | 0.5 hr | Member 4 | June 17th | June 17th | Completed |
| AT:6:1 | Run the espresso test and it passed. | 0.2 hr | Member 4 | June 17th | June 17th | Completed |
| AT:6:1 | Code documentation, to have the complete Java doc, meaningful variable and method names, inline comments where necessary. | 1.5 hr | Member 4 | June 17th | June 17th | Completed |
| AT:7:1 | Write Espresso tests | 2h | Member 2 | June 14th | June 14th | Completed |
| AT:7:1 | Update logic to use unique ID for medication | 1h | Member 2 | June 14th | June 14th | Completed |
| AT:7:1 | Refactor to separate business/UI logic | 2h | Member 2 | June 14th | June 17th | Completed |
| AT:7:1 | Implemented details page for medications | 4h | Member 5 | June 11th | June 11th | Completed |
| AT:11:1 | Build UI to populate list of medication from database | 1h | Member 3 | June 8th | June 11th | Completed |
| AT:11:1 | Ensure list is updated dynamically | 3h | Member 3 | June 8th | June 11th | Completed |
| AT:11:1 | Update Espresso tests from US-5 | 1h | Member 3 | June 11th | June 11th | Completed |
| AT:13:1 | Ensure field is properly added with unit test | 30 minutes | Member 3 | June 8th | June 9th | Completed |
| AT:13:1 | Update UI and database to accept new field | 1h | Member 3 | June 8th | June 9th | Completed |

**3. Work Schedule (January 18th 2022 ~ February 13th 2022)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| **17** | **18** | **19** | **20** | **21** | **22** | **23** |
|  |  | Setup complete: Member 2 created Gitlab and Firebase repo, connected the project and pushed it onto Gitlab | **Member 1, 2, 3 & 8 and created a list of user-stories** | **Planning meeting and Client meeting to discuss user story selection** | **Resolved firebase setup issues** | Member 8 created poll to assign people to designed user stories |
|  |  |  |  |  |  |  |
| **24** | **25** | **26** | **27** | **28** | **29** | **30** |
|  |  | **User story 2 group meeting: Members 2, 4 & 8 discussed Ats and distributed the tasks** | Member 2 set up android components and architectures | Member 2 created UI | User story 2: members 2 & 4 implemented 4 AT tests | User story 2: member 8 implemented 2 AT tests |
|  |  | Member 3 set up GitLab and Member 4set up Jira |  | **User story 2 members faced and resolved issues regrading expresso test setup** | **User story 2: member 4 implemented email validation using RegEx** |  |
| **31** | **1** | **2** | **3** | **4** | **5** | **6** |
|  |  |  |  | User story 2 meeting: discussed issues and fixes on AT4 test. |  | **User story 1 and member 8 meeting: discusses progress and plans to integrate merges into main branch** |
|  |  |  |  | **Member 8: refactored code and implemented production code for AT4** |  |  |
|  |  |  |  |  |  |  |
| **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| **Member 1 added iteration 1 document to git repository** |  |  |  | **Client Meeting: dropped user story 4 and added CI/CD pipeline assets** |  | **Member 8 worked on iteration document** |
| User story 1 & user story 2 members discussed merges | Peer review for user stories 6, 7, and 1; before pushing codes to GitLab repo. |  |  | **Fixed failing acceptance tests** |  |  |
| Member 8: attempted to refactor and merged US1 with US2 & US3. Pushed code to test branch for US1 to work on |  |  |  |  |  | Submit project report for iteration 1 |
|  |  |  |  |  |  |  |

**4. Iteration Plan Change Log [Depends on Situation if you have made any changes in the iteration plan ]**

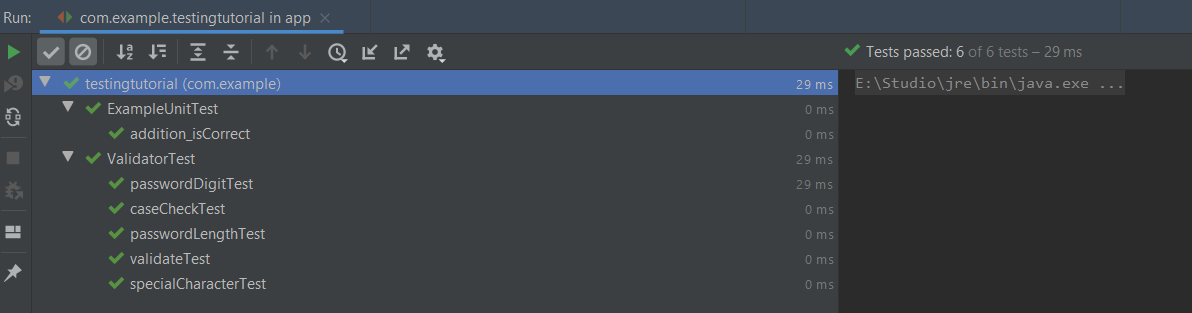
After the first week of iteration 1, we had held a meeting with the Client and discussed what user stories would be most appropriate to work on for this iteration. After the meeting, a voting system was setup so that each individual member could vote on what user story they would be most comfortable to work on. Depending on the responses, 3 sub-teams were created for the first three user stories.

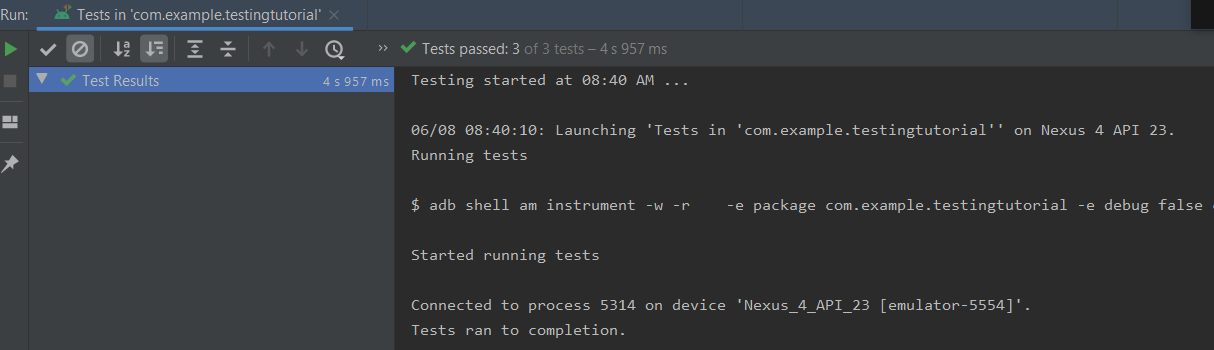
User story 2 and the merging of the first three user stories turned out to be much more difficult and time consuming than previously predicted. Hence the start of the user stories 4, 5 and 6 were delayed.

During the 3rd week of the iteration, we had to partially drop user story 6 due to lack of time and sufficient information on what to implement for it. It has been moved to iteration 2, however, the UI has been implemented for user story 6.

**Progress Charts**

1. **Testing Report**



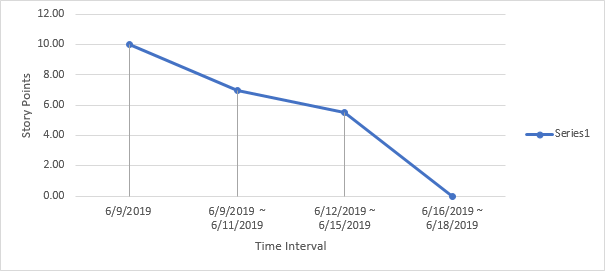


**Following are situations to consider.**

* + If you have your CI running with Junit tests, provide screenshots of Espresso tests as shown above and discussed in the tutorial.
  + If you have not yet configured CI, show the proof of testing for both Junit and Espresso test as discussed in the tutorial.

**Note**: Take snapshots of the espresso and Junit tests if not configured by the CI when every 3-4 issues are completed.

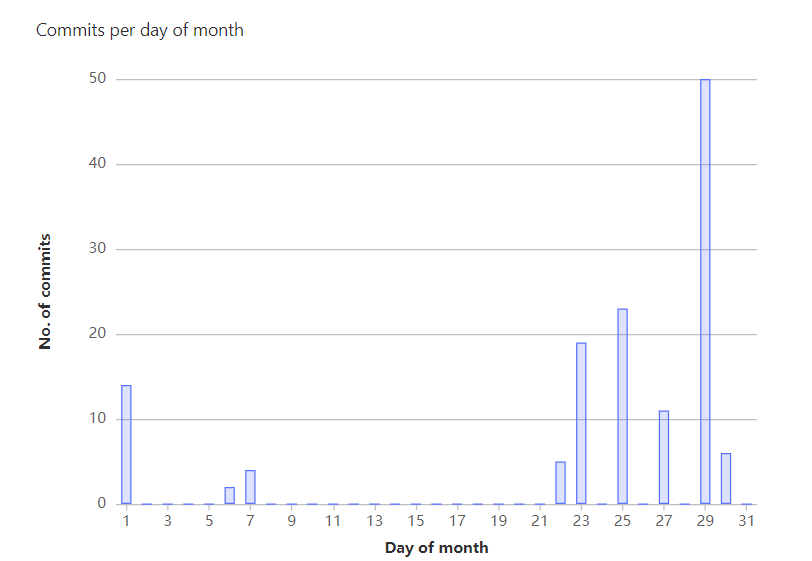
**2. Burndown Chart**

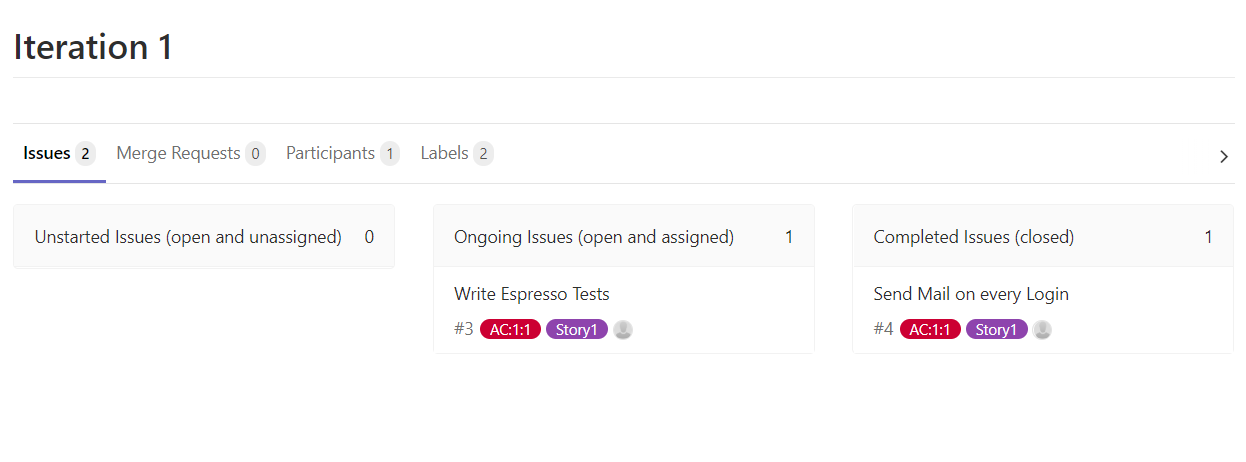


Check out the tutorial and learn how to make the burndown chart.

**3. Velocity Chart**

Check out the tutorial for more information. Sample snapshots are provided below.





**Regular Standup Meetings [Can have your own format]**

1. **General Meeting Minutes: 2019-06-03**

Time and location: Collaborate Ultra/Teams/Discord

Attendance: Members who attended

Summary:

* Iteration Planning (2 hours)
* We brainstormed, listed user stories on a white board.
* Finished User Stories cards
* We examined user stories that we currently have by giving each user story a level of confidence and some points.
* Confidence 1-5 (least to most)
* Points: 1pt = 3 hours
* Better to overestimate than underestimate time
* US-2 Logout: Make sure data is saved -> saved on firebase
* Update every time we hit save
* Logout: We have to keep in mind to force log out the user of google (push notifications)
* Update database live for functionality
* We brainstormed how we would demonstrate that the user story has been implemented, and specified acceptance tests for each user story (with storyboards, and the help of Trello).
* For each task, we considered the relationships between tasks and defined the scope of implementation.
* We also introduced our background, figured out the pair-programming plan, and how to rotate pairs each Wednesday.

**2. Lab Meeting Minutes: 2019-06-05**

Time and location: Collaborate Ultra/Teams/Discord

Attendance: Members who attended

Summary:

* Met up with client and chose the critical, significant, and optional user stories.
* Considered relations between tasks, and which tasks are of high priority in terms of implementation.
* Selected the user stories to be worked on iteration 1: US-1, US-2, US-3, US-5, US-6, US-7, US-11, and US-13
* A total of 30 points for iteration 1
* We set up the GitLab repository
* Set up the Jira with the user stories and tasks for iteration 1
* Assigned pairs and user stories for each pair
* Build a schedule

**3. General Meeting Minutes: 2019-06-10**

Time and location: Collaborate Ultra/Teams/Discord

Attendance: Members who attended

Summary:

* Each pair started working on the allocated user stories.
* We discussed about the file naming conventions, version control methods, and how to do peer review for the work.
* We ran unit tests and espresso tests for the code and followed the TDD development.
* We discussed about the engineering tests that we used for each acceptance test.

**4. General Meeting Minutes: 2019-06-11**

Time and location: Collaborate Ultra/Teams/Discord

Attendance: Members who attended

Summary:

* We started working on the allocated user stories.
* We discussed about the file naming conventions, version control methods, and how to do peer review for the work.
* We ran unit tests and espresso tests for the code, and followed the TDD development.
* We discussed about the engineering tests that we used for each acceptance test.

**5. General Meeting Minutes: 2019-06-12**

Time and location: Collaborate Ultra/Teams/Discord

Attendance: Members who attended

Summary:

* **What did each pair accomplish?**
* **Member 1 and Member2 (4 hours)**
* [US 7, not completed yet, still need espresso test]: display details for each medication, change the fields as well.
* Not using correct key for each medication. The user can also modify that. They called a query on the temporary id, they fetched the firebase id to get the data. When they created any doc in the database, they wanted to add the firebase id.
* They need to figure out what espresso tests to do.
* **Member 1 and Member2 (4.5 hours)**
* [US 1, unit tests are working, for espresso tests, only need to do something similar to assignment 2]
* **Member 1 and Member2 (6 hours)**
* [US 5 and 11]
* Added medication to database and we were able to view the name of the medication
* Espresso test focuses on the change of UML once one medication is added.
* We still need to implement the delete feature.

**Iteration Post-mortem Review [4/100]**

*Issues With Development*

*Solutions*

*Goals for Upcoming Iteration*

* *Change registration functionality to only allow user to select from Employer and Employee options instead of allowing the user to enter in a String that will only accept Employer or Employee of any casing.*

**Links**