AnyMemo-badpoem Release 2 Summary

Team members

Name	id	Number of story points and ideal hours that member was an **author** on.
Brendan McGarry	@brendanmcgarry	
Paul Richard	@prich28	33 points
Dylan Fernandes	@dylanfernandes	46 points
Hiu Tung Lam (Emily)	@Ereimei	54 points
Adam Galati	@aagalati	28 points
Matthew Teolis	@MatthewTeolis	33 points
Olivier Nourry	@ONourry	28 points

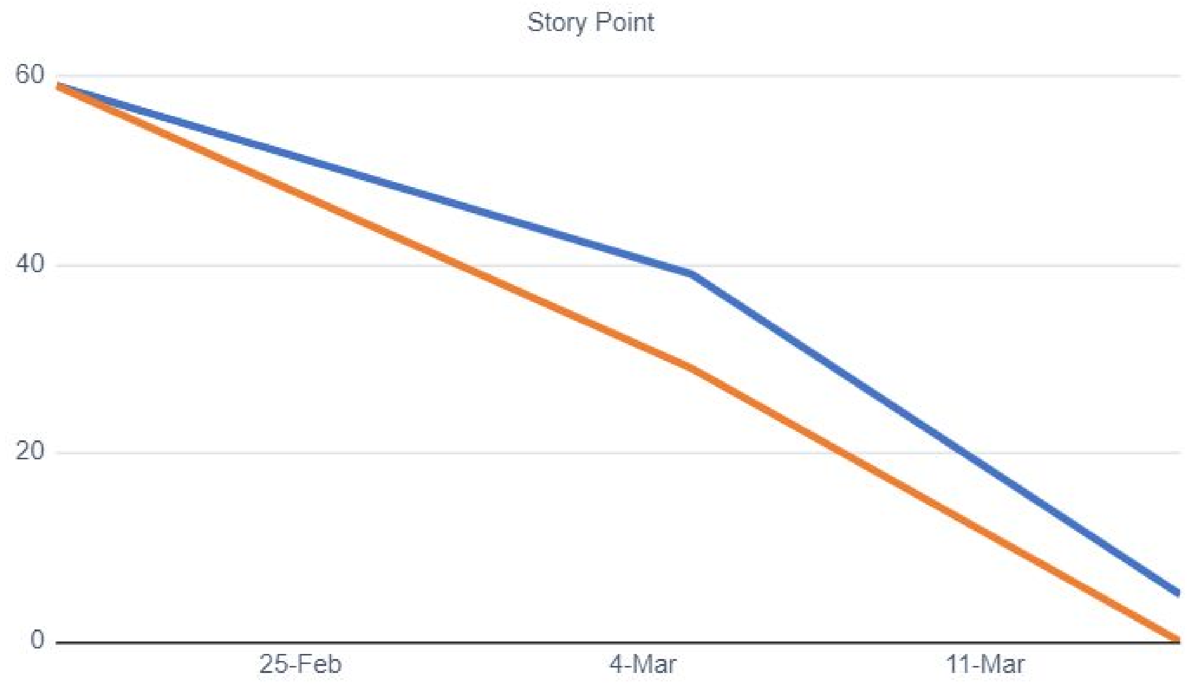
Each group member is responsible for counting their own story points. It is the group leader's duty and responsibility to make they are accurate. The team lead should be listed first.

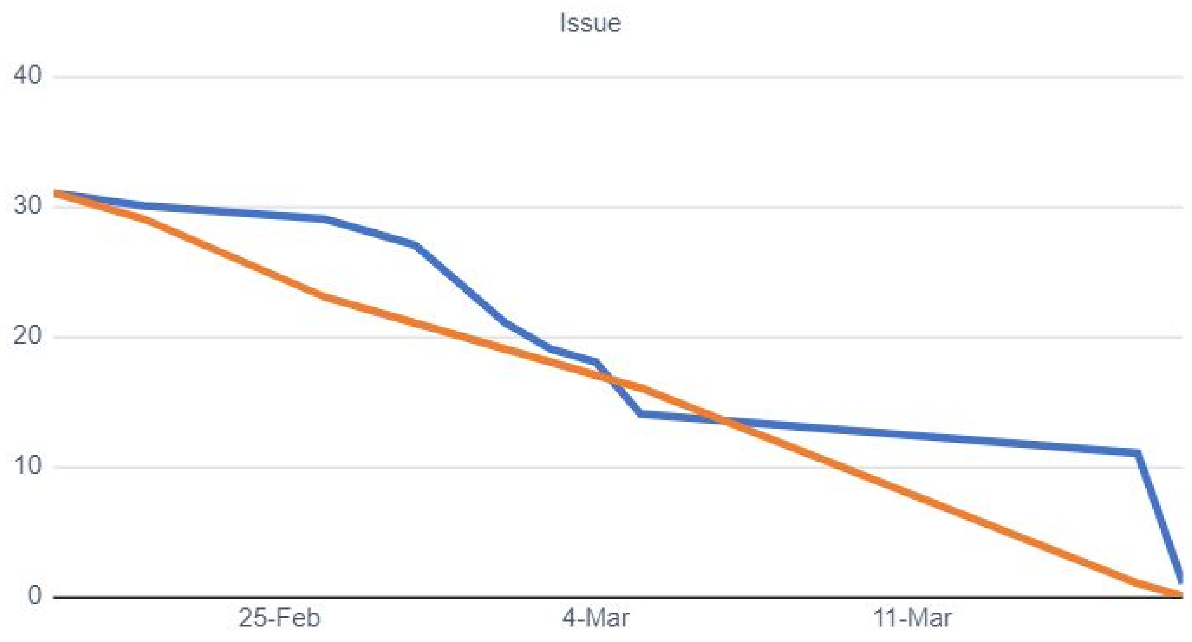
Table of Content

Velocity

How many stories did you complete over how many weeks
Total: 4 stories, 54 points, over 4 weeks
Sprint 3 (20 points)
Sprint 4 (34 points)

Burndown charts

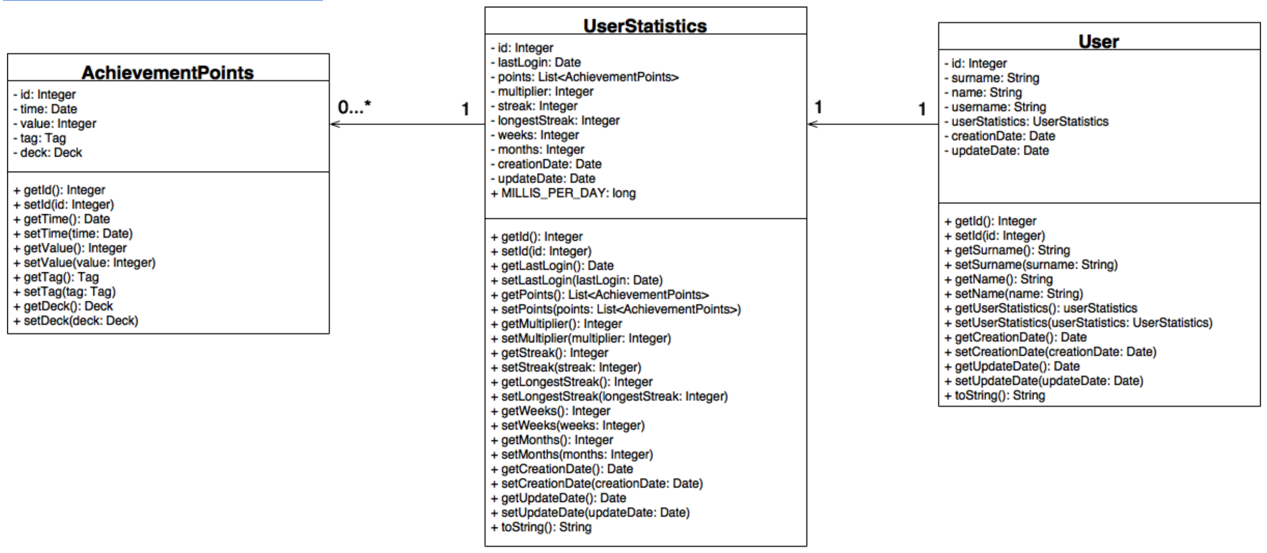




Overall Arch and Class diagram

Please describe any changes in your arch!

[Gamification document]



Plan up to next release

Just give us the links to your iterations and the total number of story points for each milestone on GitHub.

Total: 5 stories, 39 points, over 4 weeks
Sprint 5 (21 points)
Sprint 6 (18 points)

Infrastructure

No changes to infrastructure.

Name Conventions

Classes: Classes are mixed case with the first letter upper case. If the class consists of more than one word, every subsequent word follows the same convention (first letter capital). Classes names are nouns.

Interfaces: Interfaces are named like classes, mixed case with the first letter upper case. If the class consists of more than one word, every subsequent word follows the same convention (first letter capital). Interface names are nouns.

Methods/Functions: Methods and functions are mixed case with the first letter lower case. The name of a method or function is indicative of an action.

Constants: Constants are all uppercase letters. Each word is separated by an underscore ("_").

Variables: If one word, the variable is all lowercase. If more than one word, the variable is mixed case with the first word being lowercase and the subsequent words uppercase. Only iteration variables tracking a count are one letter.

Brackets and Spacing: All opening curly brackets for functions and classes are spaced one space away (on the same line) from rest of the statement. The closing bracket is placed on a seperate line after the last statement of the class or function. All indentations are a full tab from the previous indentation.

Code

Key files: top **5** most important files (full path). We will also be randomly checking the code quality of files. Please let us know if there are parts of the system that are stubs or are a prototype so we grade these accordingly.

File path with clickable GitHub link	Purpose (1 line description)

SOEN390/app/src/main/java/org/liberty/android/fantastischmemo/common/AnyMemoBaseDBOpenHelper.java	This file provides a helper for central database DAOs. It is crucial for the interaction with the central DB. Which we implemented.
SOEN390/app/src/main/java/org/liberty/android/fantastischmemo/ui/FilterActivity.java	This file is allows for the filtering of decks by tags. The user can find a deck faster searching this way.
SOEN390/blob/master/app/src/main/java/org/liberty/android/fantastischmemo/ui/TagsActivity.java	This file allows users to add tags to a deck.
SOEN390/app/src/main/java/org/liberty/android/fantastischmemo/entity/AchievementPoint.java	The AchievmentPoint entity creates a point object which can be awarded for various gamification purposes and be used to track user progress in the UserStatistics class.
SOEN390/blob/master/app/src/main/java/org/liberty/android/fantastischmemo/entity/UserStatistics.java	The class has the logic for point streaks used for gamification purposes.

Testing

Each story needs at least a tests before it is complete.
If some class/methods are missing unit tests, please describe why and how you are checking their quality. Please describe any unusually aspects of your testing approach.

List the **3** most important **unit test**s with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
SOEN390/app/src/androidTest/java/org/liberty/android/fantastischmemo/test/db/UserStatisticsDaoTest.java	Testing new UserStatistics Dao. Table creation in central DB.
SOEN390/app/src/androidTest/java/org/liberty/android/fantastischmemo/test/entity/TagTest.java	Tag Entity tests important for the completion of the TagsForDecks User Story
SOEN390/app/src/androidTest/java/org/liberty/android/fantastischmemo/test/db/TagDaoTest.java	Testing new TagDao. Table creation in central DB.

List the **3** most important **UI end to end (espresso) tests** with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
SOEN390/app/src/androidTest/java/org/liberty/android/fantastischmemo/test/ui/CreateTagInDeckUITest.java	Espresso test for creating tag in deck.

Finished SHORT Story summaries

Order your summaries by risk and priority. They should have the following form.

Points: 20, Priority: 1, Risk: High
Sprint 3: Story #56: Add tags to deck
As a user I would like to be able to add tags to decks.
Feature: Tags For Decks
Summary: This feature included making a central Database where the central Tags table is placed. The feature is a hybrid structure as all of the individual deck databases still exist and the specific tags for that deck are stored in these databases. A user interface was created to add and remove tags from individual decks.

Points: 13, Priority: 2, Risk: High
Sprint 4: Story #52: Implement Achievement System
As a user I would like to have achievements points for my progress on decks or tags
Feature: Achievement System
Summary: This feature implemented User, UserStatistics and AchievementPoint. It includes the class and the dao. It uses and stores data into the new central Database.

Points: 13, Priority: 3, Risk: Medium
Sprint 4: Story #53: Filter deck
As a user, I would like to be able to filter decks by tags to view only the decks with the tags I select.
Feature: Filter Decks by Tags
Summary: This feature included the user interface and the functionality of filtering decks by tags. It also removed the unused DeckMock.

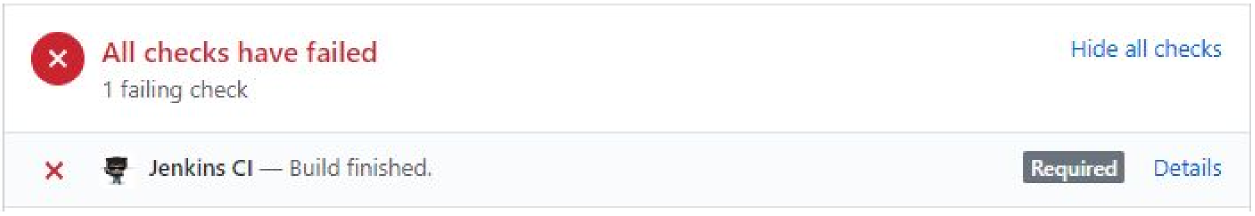
Points: 8, Priority: 4, Risk: Low
Sprint 4: Story #106: Account Page
As a user I would like to view my account information on a profile page
Feature: Account
Summary: This feature creates a default user, provides a profile page to view user's information and allows user to edit the information.

CI stages and notification

Our continuous integration pipeline works in the following way:

1. Pulling new pull requests from GitHub, either a new pull request completely or a new commit that updated the pull request
2. Assembling/compiling the code, which would check for any compiler errors.
3. Running the tests on a new emulator.
4. Report results to GitHub and Slack channel.

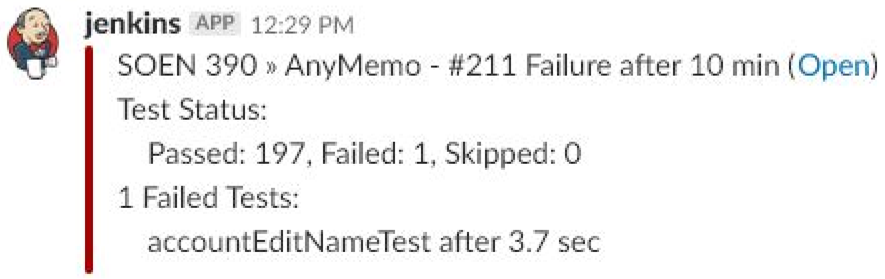
Any new pull request will trigger a new build from our Jenkins CI server by webhook. GitHub will send a URL request that will tell Jenkins what commit number to pull and test from. The next step would build the project by running the following task via gradle: "assembleFree". Once built, Jenkins will run the tests on a newly created emulator by installing the app, then it would run the tests by running the following gradle tasks, to both install the app and run the tests: "installFreeDebug" and "connectedFreeDebugAndroidTest". Lastly, once the compiling and testing is done, the results will be reported to the GitHub pull request in the following way:



All checks have failed
1 failing check
Jenkins CI — Build finished.

GitHub Status Check Notification

The results are also reported to our Slack channel, which would notify the team incase they are not looking at the pull request results at the time. The Slack notification looks like the following:



jenkins APP 12:29 PM
SOEN 390 » AnyMemo - #211 Failure after 10 min (Open)
Test Status:
Passed: 197, Failed: 1, Skipped: 0
1 Failed Tests:
accountEditNameTest after 3.7 sec

Slack Notification

Clicking "Details" or "Open" on either of the notifications will redirect you to the Jenkins server's console output. This will give a more detailed report to why the build failed, example below:

```
16:27:47 :app:connectedFreeDebugAndroidTest
16:27:47 Starting 198 tests on hudson_en-US_480_WVGA_android-21_armeabi-v7a_AnyMemo(AVD) - 5.0.2
16:28:56
16:28:56 org.liberty.android.fantastischmemo.test.ui.AccountEditNameTest >
accountEditNameTest[hudson_en-US_480_WVGA_android-21_armeabi-v7a_AnyMemo(AVD) - 5.0.2] [31mFAILED [0m
16:28:56         android.support.test.espresso.NoMatchingViewException: No views in hierarchy found matching:
(Child at position 6 in parent (with id: org.liberty.android.fantastischmemo:id/design_navigation_view and Child at
position 0 in parent with id: org.liberty.android.fantastischmemo:id/nav_view) and is displayed on the screen to the
user)
16:28:56
16:29:25 :app:connectedFreeDebugAndroidTest FAILED
16:29:25
16:29:25 FAILURE: Build failed with an exception.
```

```
16:29:25
16:29:25 * What went wrong:
16:29:25 Execution failed for task ':app:connectedFreeDebugAndroidTest'.
16:29:25 > There were failing tests. See the report at:
file:///var/jenkins_home/workspace/SOEN%20390/AnyMemo/app/build/reports/androidTests/connected/flavors/
FREE/index.html
16:29:25
16:29:25 * Try:
16:29:25 Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.
16:29:25
16:29:25 BUILD FAILED in 8m 4s
16:29:25 110 actionable tasks: 110 executed
16:29:26 Build step 'Invoke Gradle script' changed build result to FAILURE
16:29:26 Build step 'Invoke Gradle script' marked build as failure
16:29:26 [android] Stopping Android emulator
16:29:29 [android] Archiving emulator log
16:29:29 $ /var/jenkins_home/tools/android-sdk/platform-tools/adb kill-server
16:29:29 Recording test results
16:29:29 Setting status of aaf23e554cf41ff823e9a7a6541ace67d9b89b84 to FAILURE with url
http://jenkins.matthewteolis.com:8080/job/SOEN%20390/job/AnyMemo/211/ and message: 'Build finished. '
16:29:29 Using context: Jenkins CI
16:29:30 Finished: FAILURE
```

Jenkins Server Console Output

We have also created a clone of our main pipeline for testing different pipeline configs and running tests manually on commits without needing to create a new pull request. For example, we have been trying to test a specific commit over and over again by changing the pipeline configs to disable animations on the emulator in order to make tests work properly.