Team Contract

Meetings

As a project team we will:

- Book team meetings ahead of time within Taylor Library meeting spaces
- Attend team meetings, occurring at 11:30am every Tuesday in Taylor Library for a minimum of 30 minutes
- Attend TA meetings occurring at 3:30pm every Friday in MC16-A for a minimum of 20 minutes
- Not miss more than 3 meetings in a row
- Ensure meeting minutes are kept; responsibility will rotate within the group

Work Norms

We will:

- Work on project work for a minimum of 2 hours/week
- Ensure collaboration when sharing tasks
- Set and adjust team deadlines during meetings
- Review each other's work during team meetings
- Decide disagreements using majority rule. Effective as a group of 5
- Upload and update our Confluence space, and importantly main page!

Work Division

We will:

- Do our best to ensure the division of work is equal
- Be understanding of each other's schedule; demonstrate flexibility in the 'long run'
- Split future work into completable tasks
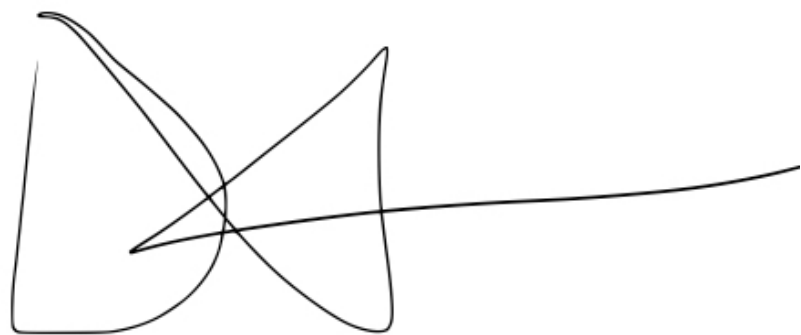- Utilize Jira on a consistent basis

Signatures

Marc

morelli

March 17 2023 - meeting

- Organize the confluence page (especially meeting minutes)
- Update team contract (correct room)
- continue working on code and GUI
- Accessibility stuff
    - be able to use both mouse and/or keyboard (arrows to scroll, tab to switch screens, etc)
    - make sure the colours are distinct (ex: for colourblind people)
        - recommends making colours darker when clicked, for example
- remove the POI list in building class, search by going through the floor then layer

March 10 2023

Our discussion revolved around these topics:-

- Using Bitbucket
    - Start as quickly as you can
    - Can ask TA for help.
- Due Date
    - The implementation and delivery component is due on the first week of April.
    - For Acceptance and Testing, we will have to show our project to the TA and basically show how each functional and non-functional requirement was fulfilled. The TA will have a checklist to evaluate our project.
        - This is also due on the first week of April (along with Implementation and Delivery). After we show our project to the TA, we have 24 hours to make final adjustments to the project and submit it (we can use our ;ate submission tickets).
- GUI Implementation
    - Use Swing or JavaFx.
    - Allowed to ask TA for help in case we encounter a problem and the solution is not directly given in the documentation.
- Programming
    - Can ask the TA for help
- Using NetBeans
    - Familiarise yourself with NetBeans as quickly as possible.

March 3 2023

Date

03 Mar 2023

Attendees

- Foster Giggie Jane Lee Matthew Morelli Daniel Bradley Wlodek Adams

Goals

- Ensure Jira tasks
- Design Doc due March 6th
- Think about implementation, know IDE and potential libraries
- JSON < org.json.simple gson . These are libraries to include in our java coding
- If we use java swing, it is a library
- Use good source code in Git!

Discussion items

Time	Item	Who	Notes

Action items

- [ ]

Jan 27 2023 - CS2212B Group 41 Minutes

3:20

- We'll be graded on consistent use of Jira, Confluence
- Need to create, submit and sign team contract (Feb 3) actually just a draft

3:35

- Meeting begins with Matthew, Jane, Mark and Foster
- Team Contract creation should be unanimous
    - When, where and how of meetings
    - Responsibility of minutes, division of labour and review of others work
    - What will we do in a disagreement! (vote?)
- 6 Late coupons! Each gives a 1-day extension, 3 max / project

3:43

- None (most?) of us know how to use Swing/JavaFX! "should not be too crazy" -Annabelle will send documentation

Feb 17 2023

- Use Jira and start assigning tasks
- Went over the requirements documentation
- Spoke about the Design Documentation
  - Don't worry about the GUI
  - Don't need to worry too much about the GUI components for the class diagram
  - Class diagrams should focus more on parameters, functions, associations, and relationships.
- Make a class UML document page on confluence and check any updates to diagrams.
- TA suggested implementing Search Function on Layer Class.
- "Each map has buildings, each building has floors, each floor has layers" - TA

Feb 15 2023 - Meeting Minutes

- Classes for our project:
  - Profile
    - String username, password
    - Boolean editor
    - Functions:
      - constructor
      - setUsername
      - setPassword
  - POI
    - String name, description
    - int coordinates int[2]
    - int buildingID, layerID, room#, floorID
    - Boolean favourite, selected(maybe in GUI)
  - Layer
  - Floor
    - POI List POI[]
    - int size {maybe standardized?}, floorID, mapID
    - Functions:
      - POI search(string POI) return 0 if not found, return POI if found
  - Building
    - Floor List Floor[]
    - int buildingID
    - Functions:
      - getFloor
      - POI search(string POI) return 0 if not found, return POI if found (importantly the POI stores which floor/building it's on, so we can navigate using the returned POI)
  - Campus
    - Building List Building[3]
    - Will contain weather API
    - Functions:
      - getBuilding
  - Weather API
  - Favourite POIs
    - favPOI List POI[]
    - Functions:
      - addPOI
  - User-created POIs

Feb 10 2023 - Meeting Minutes

- Design Documentation Requirements
    - Class diagrams with relationships
    - UI mockup does not need to be interactable
    - 1 Wireframe per screen that we will have
    - How will we store data? (POIs)
    - database strongly discouraged (extra work, no bonus marks)
        - we will not do this!
    - Development Environment
        - Netbeans suggested (built-in drag and drop)
        - What IDE, other tools, and external libraries will we use
    - Software must be:
        - Java desktop application
        - Javadoc
    - Patterns (inheritance, etc)
        - List at least 3
        - show them in class diagrams

- Next meeting Tuesday after class: Feb 14th
    - read design documentation before

Feb 7 2023 Meeting notes

Date

07 Feb 2023 11:30 am

Attendees

- Foster Giggie
- Jane Lee
- Marc Alex Crasto
- Matthew Morelli
- Daniel Bradley Wlodek Adams

Goals

- Ensure requirements documentation is complete for submission
- Review team contract

Discussion items

Time	Item	Who	Notes
11:30am	Ensure all content is present, correct	Everyone	- A few tweaks should be made before submission, final diagram included
11:50am	Explore how to export pages	Everyone	

Action items

☐

Feb 3 2023 3:30-3:50

- Discussed the requirements of the team contract
    - Won't be graded now, just has to be uploaded (will be graded later)
- Discussed requirements for the diagrams
- Talked about bit bucket issues
    - initialized bit bucket repository
    - Don't send files through git/bitbucket, use it only for code

Jan 31 2023 11:30am - 12:00pm

1) Discussing the team contract. Made a new Confluence page for Team Contract.

2) Marc and Foster discuss the activity diagram format for functional requirements.

3) Familiarising ourselves with Jira.

4) Working on the main page for the requirements documentation. The team should regularly update that page to show what changes have been made to the document.

5) Discussing the functional requirements (use cases, use case diagrams).

6) **Non-functional requirements** is being **done by Jane Lee**.

7) The **introduction** is being **done by Matthew**.

8) **Domain analysis** is being **done by Daniel**.

9) Summary will be done last.

10) Next meeting will be friday. By then, we should have most of our work up on Confluence.

Jan 29 2023 - 12:00pm - 1:00pm

Requirements Documentation Notes:

- Functional Requirements:
  - Actors (Daniel will do tonight):
    - User
    - Admin
    - WeatherAPI
  - Use cases to be done by: Daniel, Matthew, Jane

  - Diagrams to be done by: Foster, Marc
- Extra Functionality:
  - Current Weather, Secure user data

Team Contract Notes (things to include):

- Summarize changes on confluence documents when done (bottom right "What did you change?" textbox)
- Show up to meetings as often as you can

Extra Notes:

- Try to keep the format of the diagrams/tables consistent for everyone
- Next Meeting: Tuesday at 11:30 am

By Tonight:

- At least one use case
- At least one actor diagram

By Tuesday:

- Most use cases done
- Actor diagrams done