**CS3354 Software Engineering Final Project Deliverable 1**

**University Group Project Manager**

**Team Members:**

Ryan Vowell

Akhil Indur

Kevin Tran

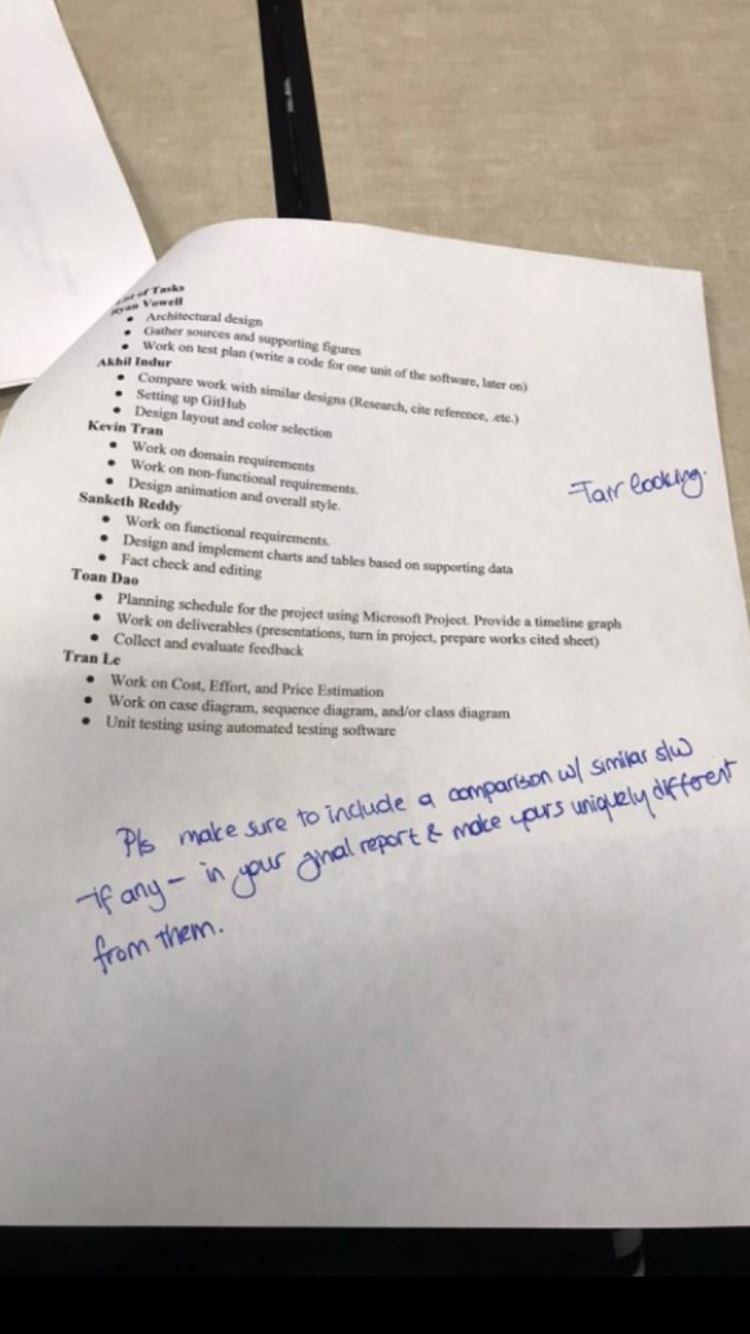
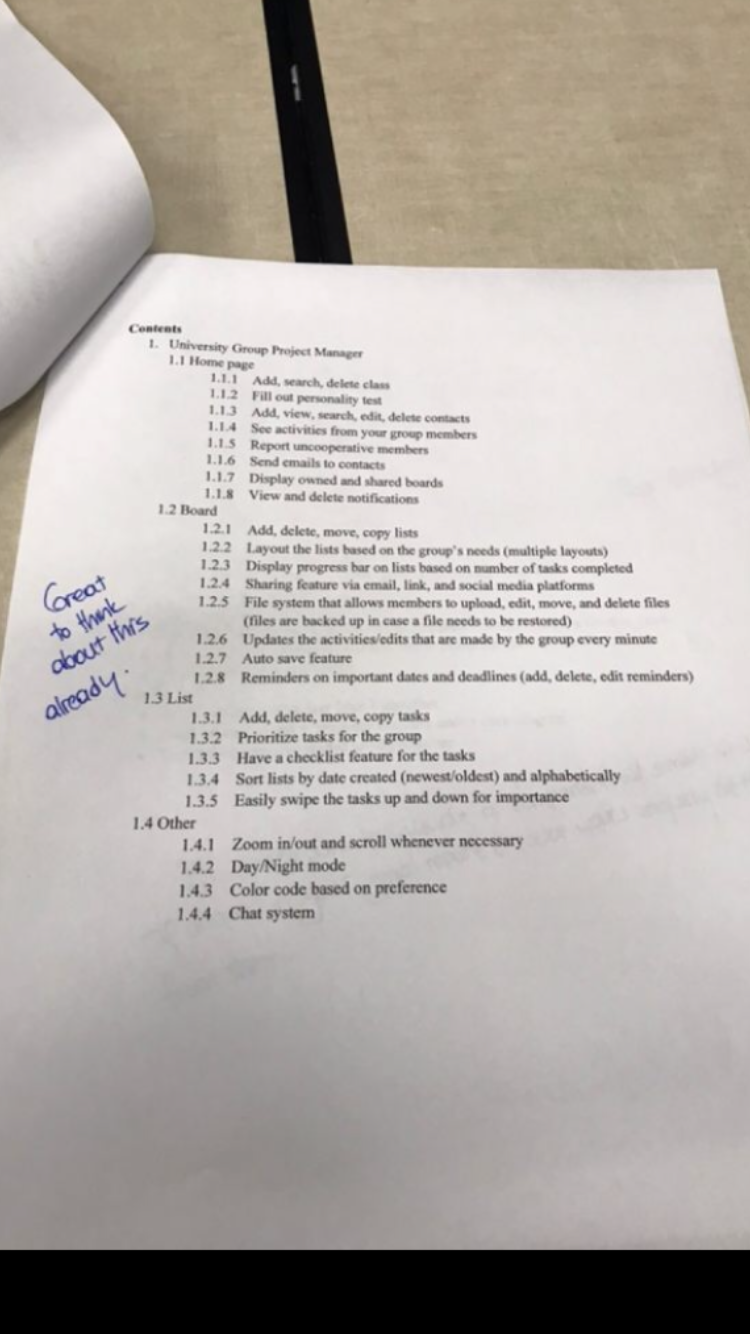
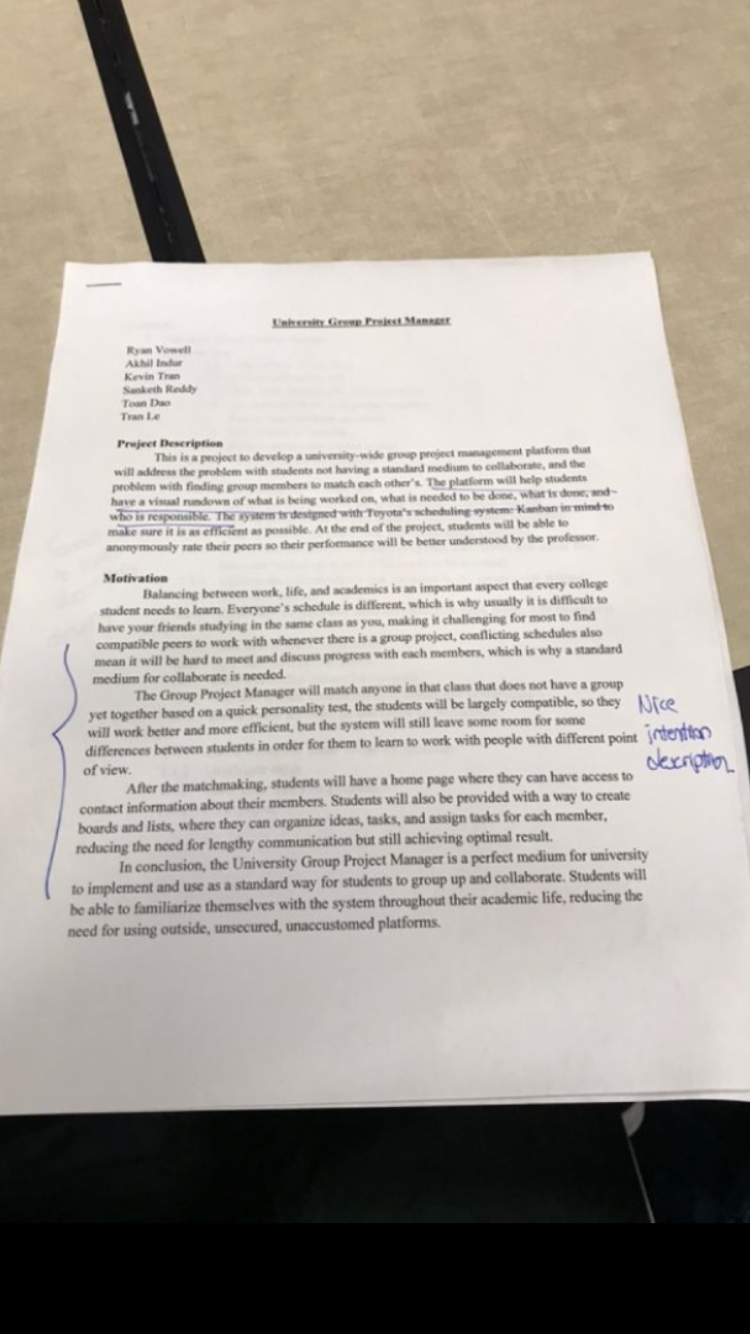
Sanketh Reddy

Toan Dao

Tran Le

**1. [5 POINTS]** Please attach here the Final Project draft description (that contains the instructor feedback). It is ok to include a picture of the original document. Address the feedback provided for your proposal by listing what you are did to comply with the proposed changes/requests for additions to your project.

**Pictures of the Final Project Draft Description that contains the instructor feedback:**



The main feedback provided by the instructor was to include a comparison with similar software and to make our software different from them. From the suggestion, the software that is similar to our University Group Project Manager is Trello. Since Trello is used for group project management, our software also provides functions for task management.

Our software and Trello both have the features of creating task boards that separate activities into groups such as “to do”, “completing”, and “done”. Our software is uniquely different from Trello in such, University Group Project Manager can be used in a closed environment only, providing only to university email accounts as our software can only be used by university students. Also, our software gives out personality tests for users to take, and our software will form groups by matching the students together based on the results of each of the personality tests. Our software also offers a communication system for each of the group members in a project.

**1. [10 POINTS]** Setting up a Github repository: 1.1. Each team member should create a GitHub account if you don’t already have one. 1.2. Create a GitHub repository named 3354-teamName. (whatever your team name will be). 1.3. Add all team members, and the TA as collaborators. Here is the TA info: TA GitHub id: xywangutd TA email: xxw140330@utdallas.edu 1.4. Make the first commit to the repository (i.e., a README file with [team name] as its content). 1.5. Make another commit including a pdf/txt/doc file named “project\_scope”. If you choose a predefined topic (one of the 4 topics described in the “Project Topic Ideas” section of this document), the contents of the file should be identical to the corresponding project in this section. If you choose other topics, the contents should follow a similar structure. 1.6. Keep all your project related files in your repository as we will check them. Include the URL of your team project repository into your project deliverable 1 report. **Important Note:**

• Tasks 1.3 - 1.5 should be performed by different team members. We will check the commit history for these activities.

• Do not include credentials (e.g., UTD ID) in the repository.

• Only commits performed before the deadline will be considered. Do not forget to push your changes after you have done the work!

**URL of team project repository:** [**https://github.com/TranLeUtd/3354-UniversityProjectManager**](https://github.com/TranLeUtd/3354-UniversityProjectManager)

**2. [5 POINTS]** Delegation of tasks: Who is doing what

**Ryan Vowell**

* Architectural design
* Gather sources and supporting figures

**Akhil Indur**

* Compare work with similar designs (Research, cite reference, etc.)
* Work on Cost, Effort, and Price Estimation

**Kevin Tran**

* Work on domain requirements
* Design animation and overall style.
* Design and implement charts and tables based on supporting data

**Sanketh Reddy**

* Work on non-functional requirements.
* Work on functional requirements.
* Fact check and editing

**Toan Dao**

* Work on deliverables (presentations, turn in project, prepare works cited sheet)
* Collect and evaluate feedback

**Tran Le**

* Setting up Github
* Work on case diagram, sequence diagram, and/or class diagram

**3. [5 POINTS]** Which software process model is employed in the project and why.

(Ch 2)

The software process that is going to be used for this project is Prototyping. The reason for the choice is because our project will be made to have basic functionalities and features first, which is going to be the first working prototype. Thee first prototype will include basic features for a management software like create tasks, projects, lists, etc.After the initial prototype and release version, we will then branch out to develop more prototypes, with each prototype be used to implement and test additional features like personal information sharing, chatting, file storage, version control, etc.

By using Prototyping, we will quickly have a basic working model to demonstrate our project and to inspire confidence in customers and group members.

**4. [15 POINTS]** Software Requirements including

**4.a.) [5 POINTS]** Functional requirements. To simplify your design, please keep your functional requirements in the range minimum 5 (five) to maximum 7 (seven). (Ch 4)

1. A user (student) shall be able to take a quick personality test.
2. The software shall match and form groups basing on the results of each student’s personality test.
3. The software shall have a homepage that displays user’s profile and overview of user’s group activities and members’ contribution.
4. Users shall be able to manage tasks between to-do list, completing list, and done list.
5. Users shall be able to create new groups, leave groups, and merge groups.
6. Users shall be able to communicate, and share documents with teammates

**4.b.) [10 POINTS]** Non-functional requirements (use **all** non-functional requirement types listed in Figure 4.3 - Ch 4)

**Product Requirement:** The system shall be available to all university students at all times.

**Usability Requirements:** Users of the system shall be only university students and every university student should be allowed to use the system

**Efficiency Requirements:** The system shall be run with small downtime.

**Performance Requirements:** The system shall be run and available to all users at all times. Downtime for the system should not be more than five seconds at any time.

**Space Requirements:** The system shall provide recommend group sizes, and each board and list shall have unlimited amounts of information in them.

**Dependability Requirements:** The system shall be available for use at all times. The maintenance of the system shall be available at all times as well.

**Security Requirements:** The system shall protect all of the students’ personal information from the outside environment.

**Organizational Requirement:** The users (students) of this system shall authenticate themselves by inputting their respective usernames and passwords which are provided by their universities and student ID’s.

**Environmental Requirements:** The system shall be used only when students enter their valid credentials when they are logging in.

**Operational Requirements:** The system shall be operated at all times.

**Development Requirements:** The system shall be developed by an experienced team, and edits in the development process shall be taking place frequently to ensure that the system is in its best possible state.

**External Requirement:** The system shall implement student privacy provisions to protect student and personal information from the outside environment.

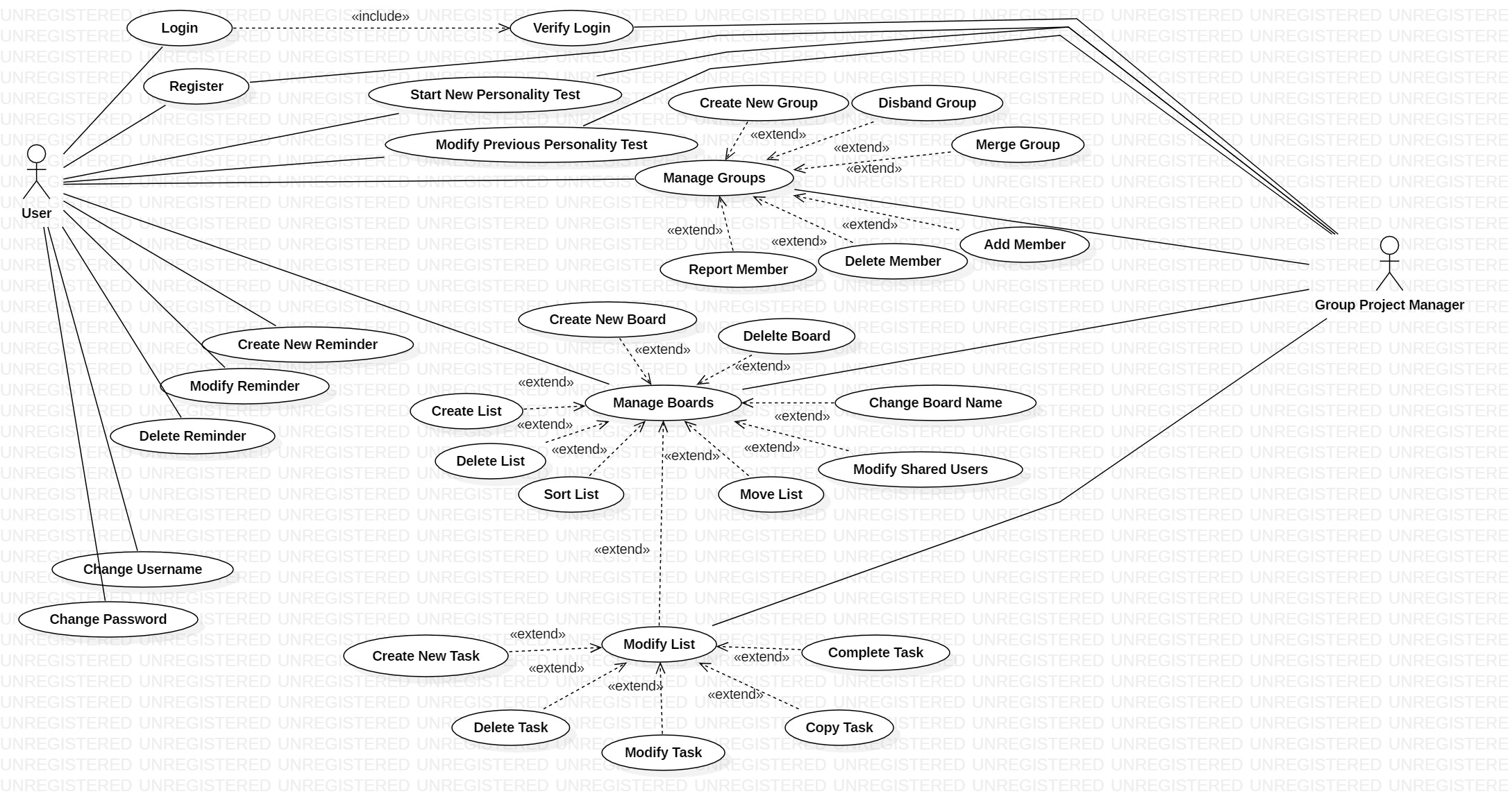
**Regulatory Requirements:** The system shall be regulated by an experienced admins and at all times to ensure that there are not any errors in the system.

**Ethical Requirements:** The system shall ensure that there will not be any ethical violations which means the system shall be confidential and protect all aspects of the students’ personal information from the outside environment.

**Legislative Requirements:** The system shall enforce rules in regards to what the students can and cannot do with the system and the data within the system.

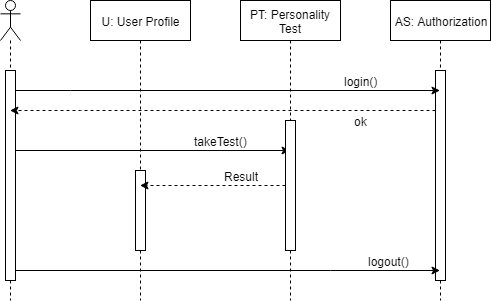
**Safety/Security Requirements:** The system shall ensure that student privacy provisions are implemented and that students’ personal information are safe secured from the outside environment.

**5. [15 POINTS]** Use case diagram – Provide a use case diagram (similar to Figure 5.5) for your project. Please note than there can be more than one use case diagrams as your project might be very comprehensive. (Ch 5 and Ch 7)

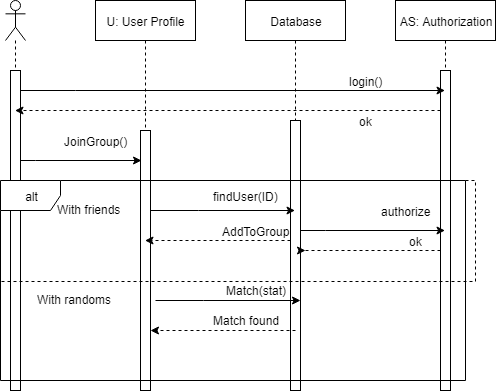


**6. [15 POINTS]** Sequence diagram – Provide sequence diagrams (similar to Figure 5.6 and Figure 5.7) for each use case of your project. Please note that there **should** be an individual sequence diagram for each use case of your project. (Ch 5 and Ch 7)

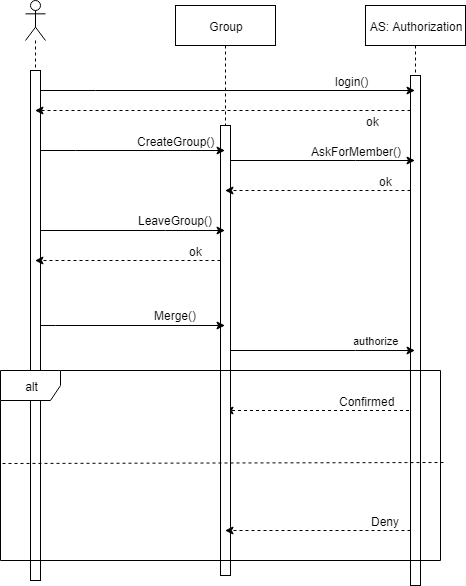
Sequence diagram for take Personality Test



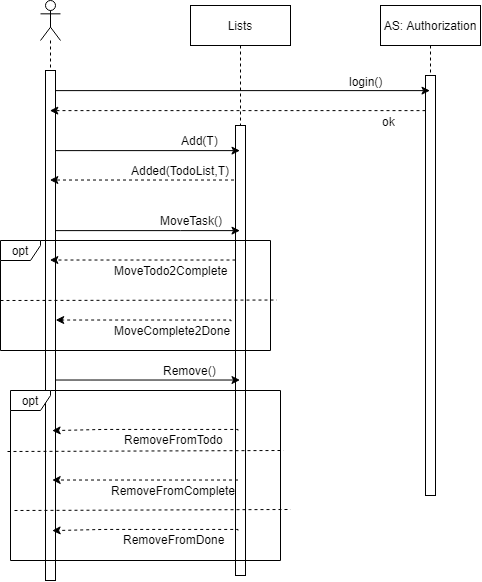
Sequence diagram for Match



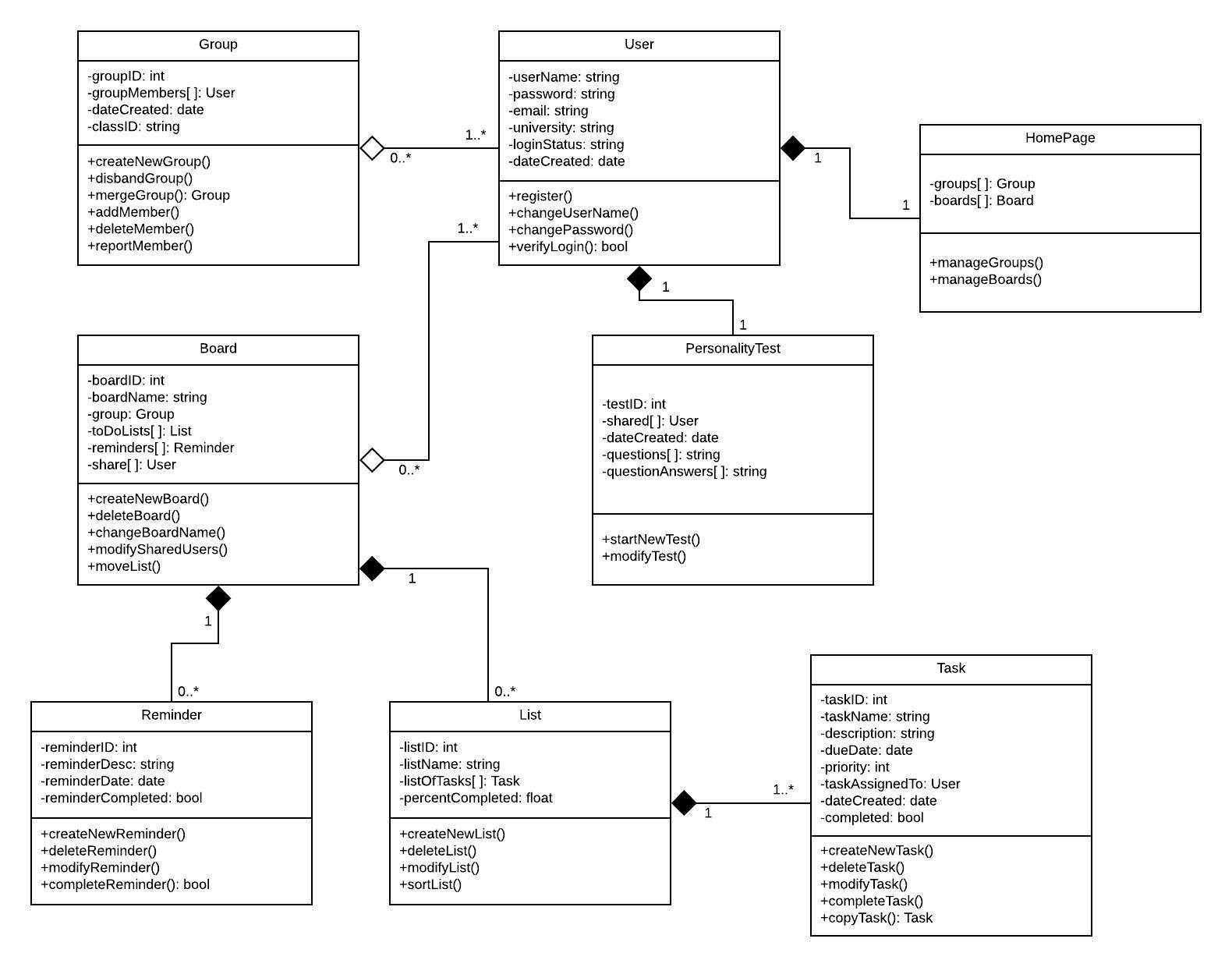
Sequence diagram for managing groups



Sequence diagram for managing lists



**7. [15 POINTS]** Class diagram – Provide a class diagram (similar to Figure 5.9) of your project. The class diagram should be unique (only one) and should include all classes of your project. Please make sure to include cardinalities, and relationship types (such as generalization and aggregation) between classes in your class diagram. Also make sure that each class has class name, attributes, and methods named (Ch 5).



**8. [15 POINTS]** Architectural design – Provide an architectural design of your

project. Based on the characteristics of your project, choose and apply **only one** appropriate architectural pattern from the following list: (Ch 6 section 6.3)

As proposed in the previous assignment, our class have decided to base our project on the Model-View-Controller (MVC).

* The Model component will be used hold user’s data like tasks, lists, texts, personal information.
* The View component will display those data in a simple and intuitive format on the web page to make it as easy to navigate and use as possible.
* The Controller will help make it easy for user to change the data from the Model component.

In conclusion, our group project architecture of choice is the Model-View-Controller pattern.

**Note that this is just a suggested outline. You are welcome to add more content when you feel necessary.**

**(NO EXTENSION IS POSSIBLE ON FINAL PROJECT DELIVERABLE 1 SUBMISSION DUE DATE). No min/max page, font type restrictions.**

**IMPORTANT NOTE: Use an automated tool for drawing all diagrams required in each deliverable. No manual drawing please.**

**Submit one copy per group.**

Please make sure the whole report is in your own words. Even if you refer to a scholar work, the words should not be exact copy and pastes, but should be rephrased in your own words. Otherwise is called plagiarism and warrants disciplinary action. Please refer to the syllabus about course policy on plagiarism.

**Project Topic Ideas:**

You may base your project idea on a fictitious institution, or an existing one. If it is an existing institution, please make sure that you comply with company privacy rules and get ALL necessary permissions from company members while obtaining and processing data.

Following is a list of some suggested topics. Some of these topics are described in more detail and others by title only. Feel free to come up with your own project topic ideas, if you want.

1. A Calendar Software

1.1 Views

1.1.1 Monthly view: show all days in a month, and event snippet for each

day 1.1.2 Weekly view: show all days in a week, and event snippet for each day 1.1.3 Daily view: show all events in a day, sorted by their starting time 1.1.4 Agenda view (optional): show all events in future as a list 1.2 Events

1.2.1 Add an event with starting and ending time 1.2.2 Check time conflicts when adding events 1.2.3 Add weekly periodical events 1.2.4 Edit & delete events 1.2.5 Event alert (optional) 1.2.6 Add/delete event categories 1.2.7 Color marking for different category of events 1.3 Share (optional)

1.3.1 Send event to other calendar users (requiring their permission) through

internet 1.4 Other

1.4.1 Holidays & weekends should be in special colors 1.4.2 Zoom in/out, and scroll support when necessary

2 SMS Messenger Software

2.1 Messages

2.1.1 View, edit, and delete messages, save a message under edition as draft 2.1.2 Send & receive messages 2.1.3 Reply & forward messages 2.1.4 Search messages by text query 2.1.5 Send to multiple receivers (optional) 2.1.6 Scheduled message (optional) 2.1.7 Auto reply (optional) 2.2 Message organization

2.2.1 Categorize messages by phone number (contact name) 2.2.2 Conversation view: view all messages between you and a certain contact,

sorted by sending/receiving time 2.3 Other

2.3.1 Zoom in/out and scroll whenever necessary

3 Contact Manager Software

3.1 Contacts

3.1.1 Add, view, edit, delete contacts 3.1.2 Support multiple phone numbers 3.1.3 Support adding a photo label for a contact 3.1.4 Search contacts by contact name 3.1.5 Blacklist (block SMS and Phones) 3.1.6 Directly make phone calls and send SMS from a contact view 3.2 Contacts Organization

3.2.1 Add contact groups 3.2.2 Manage contact groups (add/remove contacts) 3.2.3 Sort contacts by name / group name 3.3 Contacts Storage (optional)

3.3.1 Export contacts to file 3.3.2 Load contacts from file 3.4 Other

3.4.1 Zoom in/out and scroll whenever necessary

4 Book Shelf Software

4.1 Book management

4.1.1 Load books from Download folder, provide support to .txt and .pdf 4.1.2 Delete books 4.1.3 Add category of books 4.1.4 Manage categories (add/remove books) 4.1.5 Search books by text query 4.2 Book reading

4.2.1 Swipe to go to the next/previous page 4.2.2 Bookmark a page and go to the bookmark page (optional for .pdf) 4.2.3 Day & night mode (optional for .pdf) 4.2.4 Search for word and go to the word (optional for .pdf) 4.2.5 Change font and size of the text in the book (optional) 4.2.6 Extract chapters and directly go to certain chapters (optional) 4.3 Book notations (optional)

4.3.1 Add notation to certain page 4.3.2 View notations on page with notations 4.3.3 Edit notations 4.3.4 View all notations for a book 4.3.5 Delete notations 4.4 Other

4.4.1 Zoom in/out and scroll whenever necessary

5 A shipping software 6 A match making software (matching people to books/hobbies they like)

7 A comprehensive smart phone application

8 A scheduling software (an airline flight scheduler, course scheduler, task scheduler,

etc.)

9 Ticketing software such as in transportation (airline, train, cruise, etc.) domain or in

social domain (culture center, athletic complex, hotel, etc.)

10 E-commerce software

11 Online banking software

12 Non-for profit organization automation (e.g. library) software

13 University Information System software (student registration, faculty, course

schedules, ...)

14 A rental facility software involving one or more of the following: vehicles, video,

audio, books, games, and others)

15 A mobile application for suggesting a good match (for a restaurant, bookstore,

healthcare provider, etc.) based on user preferences 16 ...

This is not a comprehensive list. We encourage students to brain storm and come up with original ideas other than what is listed below. We want everyone to enjoy this semester long project and contribute the most. Therefore, think broadly and choose a topic you will enjoy working on.

Some useful links are as follows:

**UML Editors:**

Following is a list of some freely available UML editors for your convenience:

• Sparks Enterprise Architect http://www.sparxsystems.com/. 30 day trial version only.

• Violet UML editor http://alexdp.free.fr/violetumleditor/page.php. Very simple features of UML design. Free.

• Omondo EclipseUML http://www.omondo.com/ (Academic License available for free) works with Eclipse http://www.eclipse.org/

• StarUML http://staruml.sourceforge.net/en/ see also StarUML @ Wikipedia Open-source UML modeling tool supports most of the diagram types specified in UML 2.0

• UMLet http://www.umlet.com/ Open-source UML tool; runs stand-alone or as an Eclipse plug-in on Windows, OS X, and Linux

• Visual Paradigm for UML (Community Edition) http://www.visual- paradigm.com/product/vpuml/editions/community.jsp The Community Edition is free for non-commercial use; It puts a “Community

Edition” watermark on your diagrams; Runs on Windows XP/Vista/7, Linux, Mac OS X, etc.

• Netbeans UML Plug-in http://www.netbeans.org/features/uml/ Does not support all UML diagram types, but supports forward and reverse engineering

• ArgoUML http://argouml.tigris.org/ see also ArgoUML @ Wikipedia

• Rational Rose http://www.rational.com/tryit/index.jsp

• http://www.microgold.com/

• Microsoft Visio and open-source Dia are diagramming tools with a library of UML shapes that may also be used for drawing UML diagrams.

• Creatly http://creately.com/ for drawing UML diagrams.

**Making life easy when working as a group:**

It is very important to make sure that you communicate and share common work with your teammates. Here are some URLs to help you on that:

• Github — a web-based Git or version control repository and Internet hosting service. This is the recommended version control software for this project. If by some reason you cannot use the Github platform, you may use any of the following similar platforms for sharing your project related material.

• Doodle—a tool for time management and meeting scheduling.

• GroupMe—a group messaging service that lets you be in touch with your team members via mobile phones.

• CVS, open source version control - helps you work on different versions of the same product and merge your versions.

• Slack — a web-based team communication service.

• Mercurial https://www.mercurial-scm.org/ for version control.