CS3354 Software Engineering
Fall 2023
Final Project Specifications

Following is a detailed description of the project requirements for this course. Students should work in a group of **exactly 7 students** for the final project. Remember that group work is a course requirement. Each group will decide on the topic they want to work on by themselves. We will not assign topics to groups. We will dedicate end of class time to help students with forming groups, and deciding on a project topic until the proposals are due.

The important dates for the project are:

• Friday 9/15 by 11:00pm (Proposals will NOT be graded. It is posted to collect information about members of a group, make sure each student is part of a group, and to see what each group will work on. We will provide feedback to each group's proposal. Since eLearning won't post an assessment without a grade, you will see a 0 (zero) grade on eLearning for the proposals. Do not panic. We won't use that zero for anything: Submission of Final Project proposal. Please designate ONE MEMBER of your team to submit your proposal via eLearning. There will be a Final Project Proposal Assessment created on eLearning for you to submit your group proposals. **One submission per group please. So, designate one group member to submit all group related documentation to eLearning. Doing otherwise will cause a lot of confusion and delay. Please DO NOT EMAIL anything, only eLearning submissions.**

- The title of your project
- the group members (firstname and lastname)
- what you will be doing
- a detailed description of your motivation (why you chose to do this particular project), where you expect your design to be used in real life.
- the list of tasks delegated to each member: the task delegation should be distributing what to do from start to completion of the project to each member of your group. It does not have to be too detailed. One sentence for each member is sufficient.
- are you interested in writing a scholar paper in the end (no extra grade advantage)

• Friday 10/20 by 11:00 pm: (50% of your final project grade) Submission of Final Project deliverable 1. Please designate ONE MEMBER of your team to turn in your proposal via eLearning. There will be a Final Project Proposal Assessment created on eLearning for you to submit your group proposals. **One submission per group please. So, designate one group member to submit all group related documentation to eLearning. Otherwise will cause a confusion. Please DO NOT EMAIL anything, only eLearning submissions.** Project deliverable 1 is primarily for helping you keep as planned with your project, and provide help as needed. The instructor will read your deliverables and provide feedback. So, please make sure to submit your work timely and as a soft copy, as requested. One deliverable per group please.

Project deliverable 1 is primarily for helping you keep as planned with your project, and provide help as needed. So, please make sure to submit your work timely and as a soft copy, as requested. Deliverable 1 should include:

- **Setting up a Github repository.** Please use your utdallas email accounts only for each group member**:**
  - 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 (GitHub id will be provided once we know the TA) as collaborators.
  - 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. 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!

Also include the following into your project deliverable 1 report:

- The title of your project

  Delegation of tasks including each group member (firstname and lastname): List who is doing what. If no contribution, please specify as it will help us grade each group member fairly. You can certainly make assumptions, even make up government/country based rules, requirements to be able to provide one for each. Please explicitly specify if you are considering such assumptions.)
- An addressing to the feedback that we will have provided for the Final Project proposal. List what you are doing / planning to do regarding the feedback provided for your project proposal.
- Which software process model is employed in your project and why. (Ch 2)
- List of software requirements including
  - Functional requirements. To simplify your design, please keep your functional requirements in the range minimum 5 (five) to maximum 7 (seven). (Ch 4)

    - Non-functional requirements (use **all** non-functional requirement types listed in Figure 4.3 - Ch 4. This means for each of the leaves of Figure 4.3.)
- Use case diagram – Provide a use case diagram (similar to Figure 5.5) for your project. Please note than there **should** be more than one use cases depending on the complexity of your project. (Ch 5 and Ch 7)
- 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)
- 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).
- 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)
    - Model-View-Controller (MVC) pattern (similar to Figure 6.6)
    - Layered architecture pattern (similar to Figure 6.9)
    - Repository architecture pattern (similar to Figure 6.11)
    - Client-server architecture pattern (similar to Figure 6.13)
    - Pipe and filter architecture pattern (similar to Figure 6.15)

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

IMPORTANT NOTE: Please use an automated tool for drawing all diagrams required in the deliverables. No manual drawing please.

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.

• **Friday 11/17 by 11:00 pm: (50% of your final project grade)** Submission of Final Project deliverable 2. Please designate ONE MEMBER of your team to turn in your proposal via eLearning. There will be a Final Project Proposal Assessment created on eLearning for you to submit your group proposals. **One submission per group please. So, designate one group member to submit all group related documentation to eLearning. Otherwise will cause a confusion. Please DO NOT EMAIL anything, only eLearning submissions.**

The following is required for the Final Project Deliverable 2 submission:

- The title of your project

- The group members (firstname and lastname)
- Well described delegation of tasks, i.e. who did what in the project. Please make sure to fairly distribute tasks in the team and remember that each member of the same team will receive the same grade. If no/poor contribution by a member, please specify clearly so that we can grade each student fairly.
- Everything required and already submitted in Final Project Deliverable 1

IMPORTANT NOTE: The following items will all need to be calculated / worked on based on the project you are designing. As an example, if a team of 7 students in CS3354 class is working on the development of a hospital information system, this group will prepare the project scheduling, cost, effort and pricing estimation calculations based on the hospital information system design, NOT based on their 7 people team. Think of the analogy to the "*Inception*" movie: What you will be working on is the dream in a dream, i.e. the dream in the second level, NOT in the first level.

- Project Scheduling, Cost, Effort, and Pricing Estimation, Project duration and staffing: Include a detailed study of project scheduling, cost and pricing estimation for your project. Please include the following for scheduling and estimation studies:
  - Project Scheduling. Project Scheduling. Make an estimation on the schedule of your project. Please provide start date, end date by giving justifications about your estimation. Also provide the details for:
    - Whether weekends will be counted in your schedule or not
    - What is the number of working hours per day for the project
  - Cost, Effort and Pricing Estimation. Describe in detail which method you use to calculate the estimated cost and in turn the price for your project. Please choose one of the two alternative cost modeling techniques and apply that only:
    - Function Point (FP)
    - Application composition
  - Estimated cost of hardware products (such as servers, etc.)
  - Estimated cost of software products (such as licensed software, etc.)
  - Estimated cost of personnel (number of people to code the end product, training cost after installation)
- A test plan for your software: Describe the test plan for testing minimum one unit of your software. As an evidence, write a code for one unit (a method for example) of your software in a programming language of your choice, then use an automated testing tool (such as JUnit for a Java unit) to test your unit and present results. Clearly define what test case(s) are provided for testing purposes and what results are obtained. (Ch 8)
- Comparison of your work with similar designs. This step requires a thorough search in the field of your project domain. Please cite any references you make.
- Conclusion - Please make an evaluation of your work, describe any changes that you needed to make (if any), if things have deviated from what you had originally planned for and try to give justification for such changes.

- References: Please include properly cited references in IEEE paper referencing format. (You may see a referencing example in the sample IEEE paper in URL: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf) It means that your references should be numbered, and these numbers be properly cited in your project report.

  Also include:
- Your presentation slides
- Implemented code (the complete code, if any. Otherwise the unit required in the "A test plan for your software" section described above). Please note that an implementation is not required for your project. It is optional and if you choose to do so, your work may qualify for a potential publication as a scholar article.

  GitHub requirement:
- Make sure at least one member of your group commits everything for project deliverable 2 to your GitHub repository, i.e.
  - Your final project deliverable2 report
  - Unit test code for a sample unit of your project
  - Implementation code (if you have implemented your project)
  - Presentation slides

Still, one member of your team should also submit the required project deliverable 2 materials to eLearning.

(NO EXTENSION IS POSSIBLE ON FINAL PROJECT DELIVERABLE 2 SUBMISSION DUE DATE). No min/max page, font type restrictions.
To be fair with each student, each project should be submitted via eLearning on the same date, regardless of when the group is going to present his work. Projects will be conducted throughout the semester. So, the report and presentation should be prepared accordingly (comprehensive enough with properly cited conference and/or journal paper references).

IMPORTANT NOTE: Please use an automated tool for drawing all diagrams required in the deliverables. No manual drawing please.

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 Presentation:

Each group is expected to present their work in person during class time on one of the scheduled dates as seen in the syllabus. Each member of a group should be present and talk during presentations.

11/28
11/30
12/05
12/07

Each presentation should take **maximum 20 minutes. No min. time requirement.**

The presentation schedule will be announced once ALL groups turn in their final project deliverable 2. No particular order yet. As each group will submit their final report (deliverable 2) on the same date, there will be no advantage/disadvantage for any group in terms of presentation date. Potentially, your group could be presenting on either of the above 4 dates. Plus, each team is expected to be present and listen to other teams' presentations. Those presentations are very useful and help us learn a lot from each other.

Feel free to enrich your presentation with supporting figures, charts, documents, tables, similar work, etc.

Contribute from yourselves: Employ your own design layouts, color selections, animations, artistic perspectives to your presentations. Try to make them attractive. Think of commercials: We only remember the "interesting" ones.

While working in a group, it is your responsibility to delegate tasks fairly and ensure that everybody contributes. Please remember that you are allowed to form your own groups. **Members of the same group will receive the same grade,** unless group members report poor member performance. So, if there is any problem in your group, please act timely.

Make sure each group member talks during the presentation. You may arrange presentation times for each member based on the flow of your presentation.

It is a suggested tactic that in a presentation, each slide should remain min. 1 on display so that everybody reads and understands it. So, please plan ahead accordingly.

Rehearsal will prevent unexpected surprises. Make sure you rehearse and time your presentation before you actually present it.

Project Topic Ideas:

Each group will decide on the topic they want to work on by themselves. We will not assign topics to groups. 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 boo
  - 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.

Useful Tools:

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.