**CS3354 Software Engineering**

**Final Project Deliverable 2**

Storage Hub

Kuanlin Liu

Cameron Glick

Jesse Ladyman

Kevin Tran

Khiem Huynh

Yeswanth Bogireddy

Zareef Kabir

1. **[5 POINTS]** Well described delegation of tasks, i.e. who did what in the project. Now that your project is complete, you are required to submit the delegation of tasks from beginning of the project until the end. Please make sure to fairly distribute tasks in the team and remember that in the end of the semester, each member of a team will receive the same grade. See grading policy below for more detail.

Delegation of Tasks:

Software process model and requirements. [Kuanlin and Kevin]

Diagramming models (use case, sequence, class). [Cameron and Jesse]

Architectural design. [Khiem and Zareef]

Project scope. [Yeswanth]

1. **[5 POINTS]** Everything required and already submitted in Final Project Deliverable 1. Please specify this part as “Project Deliverable 1 content”.

Will be found as “Final Project Deliverable 1” in the zip file.

|  |  |
| --- | --- |
| **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 6 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 6 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.** |  |

1. **[50 POINTS]** 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:

* 1. **[5 POINTS]** Project Scheduling. Please note that what you present should be the timeline of the project designed, NOT the time you’ve spent on it.

You may use personal judgement to come up with a hypothetical timeline for your project. Please provide supporting facts for your assumptions.

* 1. **[30 POINTS]** Cost, Effort and Pricing Estimation. Use the Function Point algorithmic estimation method to estimate the effort and time, and in turn the price for your project.

* 1. **[5 POINTS]** Estimated cost of hardware products (such as servers, etc.)

* 1. **[5 POINTS]** Estimated cost of software products (such as licensed software, etc.)

* 1. **[5 POINTS]** Estimated cost of personnel (number of people to code the end product, training cost after installation)

Assuming we have about 6-8 employees (including one person being delegated to training others), costs can reach high rather quickly. According to *Glassdoor* the average software developer in the U.S. makes $76,526 a year, however sites such as *Fixr* and *UsNews* cite the figure as high as $93,350 and $103,620 respectively. However, *Forbes* lists the number as low as $72,210 for the salary of a software developer of an application. Given this information we can begin to estimate the cost of the salary of all the employees combined. Now if we ourselves were to actually develop a system of this sort in our current state with our current experience in a small company just starting out, we can be assumed to be achieving a salary on the lower ends of the spectrum; $72,210 was the lowest salary listed on Forbes and if were to multiply this number by 5 or 6, we’d get a value in the range of $433,260 - $577,680. In total, we’d have to pay all employees this much.

Citations:

1. “Salary: Software Developer,” *Glassdoor*. [Online]. Available: https://www.glassdoor.com/Salaries/software-developer-salary-SRCH\_KO0,18.htm. [Accessed: 19-Apr-2020].
2. DePietro, “Here's How Much Money Software Developers Earn In Every State,” *Forbes*, 21-May-2019. [Online]. Available: https://www.forbes.com/sites/andrewdepietro/2019/05/21/software-developer-salary-state/#3d548a8c2f2c. [Accessed: 19-Apr-2020].
3. “How Much Can a Software Developer Expect to Get Paid?,” *U.S. News & World Report*. [Online]. Available: https://money.usnews.com/careers/best-jobs/software-developer/salary. [Accessed: 19-Apr-2020].
4. Y. Kato, “How Much Does a Software Developer Make in Every State?,” *Fixr Blog*, 03-Dec-2019. [Online]. Available: https://www.fixr.com/blog/2015/02/20/how-much-does-a-sofware-developer-make-us/. [Accessed: 19-Apr-2020].

However, if we were just to calculate how much it may cost to develop, we may achieve a much different number. We can use the website *Thumbtack* which goes into detail into how much development may cost for a software per hour. Going off the fact that we’d likely need a combination of not just Front-end and Back-end development but also API, Desktop, and mobile app development we can hope to assume a rather large cost here as well. *Thumbtack* lists the following as the prices for development.

* Basic C development: $75-$150 per hour
* Front-end web development: $50-$75 per hour
* Back-end web development: $75-$150 per hour
* API development: $75-$150 per hour
* Desktop app development: $30-$100 per hour
* Mobile app development: $30-$150 per hour

For the purposes of our app, we can assume the costs to be closer to the lower ends of the presented ranges (as we’d be a smaller development team). We can also expect close to around 100 hours in total for each piece of development, through this, we can achieve a value of around $33,500

(75 \* 100) + (50 \* 100) + (75 \* 100) + (75 \* 100) + (30 \* 100) + (30 \* 100) = 33,500

Citation:

1. T. Editors, “How much does software development cost?,” *Thumbtack*, 26-Sep-2017. [Online]. Available: https://www.thumbtack.com/p/software-development-costs. [Accessed: 19-Apr-2020].

1. **[10 POINTS]** 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). Include your test code as additional document in your zip file submitted.

1. **[10 POINTS]** 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.

1. **[10 POINTS]** 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.

1. **[5 POINTS]** References: Please include properly cited references in IEEE paper referencing format. Please review the IEEE referencing format document at the URL:

[https://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf)](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf). It means that your references should be numbered, and these numbers properly cited in your project report.

**Also include:**

1. **[5 POINTS]** Your presentation slides. No min/max number of slides enforced. Please make sure that you can complete presentation within **12 (twelve)** minutes.

Following template could be a good start to prepare your presentations. As each project topic is different, a variety in presentation style is expected and welcome.

* + - Title of your project together with participants
    - Objective of the project designed
    - Cost estimation
    - Project timeline (timeline of the project designed, NOT the time you’ve

spent on it)

* + - Functional and non-functional requirements. If too long, select representative items. - Use case diagram
    - Sequence diagram for a selected representative operation of the project.
    - Class diagram
    - Architectural design
    - 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)
    - Preferably a demo of user interface design that shows screen to screen transitions though no full functionality is required.
    - OPTIONAL: IF implemented the project, a demo of your implementation.

1. **OPTIONAL PART.** Your program code **(if fully implemented the project, not required otherwise).** Please note that **implementation is not required for the final project**. Groups are welcome to implement their work, if they choose to do so.  **[This part may qualify for extra credit, if you implement and submit the implementation code together with your project. The extra credit will be determined based on the quality of your implementation]**

**Please note: This is just a suggested outline. You are welcome to add more content if you feel necessary.**

**(NO EXTENSION IS POSSIBLE ON FINAL PROJECT DELIVERABLE 2 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.**

**UML Editors:**

Following is a list of some freely available UML editors for your convenience: • Sparks Enterprise Architect [http://www.sparxsystems.com/.](http://www.sparxsystems.com/) 30 day trial version only.

* Violet UML editor [http://alexdp.free.fr/violetumleditor/page.php.](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](http://en.wikipedia.org/wiki/StarUML)

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](http://www.eclipse.org/) plug-in on Windows, OS X, and Linux

* Visual Paradigm for UML (Community Edition) [http://www.visualparadigm.com/product/vpuml/editions/community.jsp](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](http://en.wikipedia.org/wiki/ArgoUML)
* Rational Rose<http://www.rational.com/tryit/index.jsp>
* <http://www.microgold.com/>
* [Microsoft Visio](http://office.microsoft.com/en-us/visio/) and open-source [Dia](http://live.gnome.org/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](https://github.com/) — 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—](http://www.doodle.com/)a tool for time management and meeting scheduling.
* [GroupMe—](http://groupme.com/)a group messaging service that lets you be in touch with your team members via mobile phones.
* [CVS, open source version control](http://www.nongnu.org/cvs/)  - helps you work on different versions of the same product and merge your versions.
* [Slack](http://slack.com/) — a web-based team communication service.
* Mercurial <https://www.mercurial-scm.org/>for version control.

**About Presentation of your Project:**

|  |  |
| --- | --- |
| **Due to online transformation after the Spring Break, no live presentations before class. Instead, your voice recorded presentations will be posted to eLearning for our class to access them. Therefore, each group needs to voice record its presentation**  **slides. Though being distant shouldn’t affect recording a presentation in which each member of the team speaks, we still would like to provide flexibility, considering it might be needed due to individual requirements under the current situation. So, if you choose to, you may designate one member of your team and he/she does the entire voice recording. In any case, please submit your recorded slides along with your final project report (deliverable 2). Please make sure your presentation is max. 15 minutes. No minimum time requirement. You may use any technology for voice recording your slides. Please use this as an opportunity to explore new techniques** | |
| **and learn something new.** |  |

You may use any style in your presentations. A slide show is recommended as it helps with displaying a summary of content you talk about to the audience as well as to yourself. A suggested outline for presentations is listed below:

A brief introduction to your project topic

List of requirements

Use case diagram that contains use cases

Sequence diagram

Design Class Diagram (DCD)

User Interface Design

Comparison with similar work (if any), or emphasizing its significance and uniqueness (if there is not any)

Conclusion and Future Work

You are welcome to enhance the minimum content listed above, provided that you stay within 15 minute presentation time requirement.

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.

It is a suggested tactic that in a presentation, each slide should remain min. 1 minute on display so that everybody reads and understands it. So, not too many slides maybe a good idea to start with.

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

**What to submit?**

Please zip the following as one single file:

* Final project deliverable2 report (Please note that your deliverable2 report should also include your deliverable1 report as required in section 2 above.)
* Test code (section 4 above)
* Presentation slides (section 8 above)
* [Optional] Implementation code (section 9 above)

**Grading Policy:**

Your project grade will be the average of your report and in class presentation evaluation. So, please make sure to delegate tasks and contribute fairly.

Remember that it is part of group work responsibility to delegate tasks and ensure that everybody contributes. **Members of each group WILL receive the same grade.** So, if there is any problem in your group, please act timely.

Writing a paper at the end of your work and submitting it to a journal/conference is strongly encouraged. I will support students who want to proceed with implementation of their design and submit their work to a journal/conference.