SE 4485: Software Engineering Projects

Fall 2024

Project Management Plan

Group Number	8
Project Title	Design, Develop, and Test a Location Verification App to be Installed on the Two Smartphone Alternatives
Sponsoring Company	The Fellows Consulting Group
Sponsor(s)	Tom Hill
Students	1. Abdullah Chaudhry 2. Omar Hussain 3. Reya Dawlah 4. Danya Almintakh 5. Alvin Mathew 6. Mageto Nyakoni

ABSTRACT
This Project Management Plan outlines the framework for the successful design, development, and testing of a mobile location verification application for the Fellows Consulting Group. The document details the project's objectives, scope, lifecycle model, risk analysis, and the roles and responsibilities of the team members. It also describes the necessary software and hardware resources, the project organization, and the methods for monitoring and controlling the project's progress. Additionally, it covers professional standards and expected behaviors within the team, ensuring alignment with project goals and ethical practices. This plan provides a comprehensive roadmap for delivering a robust and reliable application that meets the sponsor's requirements.

INTRODUCTION

This document serves as the Project Management Plan for a mobile application we will create for the Fellows Consulting Group. This document outlines the objective, scope, structure of the team, and a basic outline of how the project will be executed. It highlights the team's approach on designing, developing and testing a location tracking/verification app, and presents a clear picture for the stakeholders.

Purpose and Scope of the document

The purpose of this Project Management Plan is to define the framework within which the project will be managed. This includes the project's objectives, deliverables, milestones, and the roles and responsibilities of the project team members. The scope of this document covers all phases of the project, from initial planning and requirements gathering to design, development, testing, and final delivery of the mobile application.

Brief overview of the product

The product to be developed is a mobile location verification application for Fellows Consulting Group. The main purpose of this app is to provide real-time verification of the identity and location of in-home service delivery personnel. The app will be installed on two different smartphone platforms and will be used to ensure that the person delivering services to a home is indeed the person they claim to be and that they are at the correct location. It should be capable of tracking the service personnel's location in real time and be installed on their device. This allows the homeowner to verify that it's the correct service personnel, and also for the service provider to track their employees location.

The document gives a brief introduction of the product to be developed, and then branches out into the people who will be creating it, methods and procedures followed by the team to deliver this product, as well as the tentative timeline.

PROJECT ORGANIZATION

Formed in our Software Engineering Project class, Team 8 consists of 6 people, seated under the Project Sponsor of Dr. Tom Hill. The designated group leader is Abdullah Chaudhry, which was decided upon on a volunteering-basis.
We communicate through an iMessage group chat for updates on deliverables, scheduling and submission, and any information necessary on a group level.

We have arranged to meet on a weekly basis at 4 pm to check in with our sponsor, in order to review our final submissions and get the green light to turn our work in to Professor Wong. Following our weekly 30 minute check-in, we meet virtually as a team for 40 minutes to an hour in order to formally approve submission on a team level.

Given the project requirements and traditional roles within development, we have listed 6 roles below that our team have discussed and assigned according to our pre-existing skillsets with names listed beside each role, in order to ensure speedy delivery. These roles aren't strict, however they will be used as a guideline for project responsibilities/SMEs within the team.

Core Roles:
1. **Project Manager:** Oversees the entire project, ensures deadlines are met, and manages team

dynamics. ***Abdullah***

2. **UI/UX Designer:** Creates the app's interface, user experience, and wireframes, ensuring a seamless and intuitive user journey. ***Mageto***
3. **Front-End Developer:** Builds the app's user interface and handles interactions using technologies like HTML, CSS, and JavaScript. ***Omar***
4. **Back-End Developer:** Develops the server-side logic and API for the app, handling data storage, retrieval, and location verification processes. ***Abdullah, Alvin***

Supporting Roles:
1. **Quality Assurance Engineer (QA):** Tests the app throughout development, identifies bugs, and ensures it meets quality standards. ***Danya***
2. **Mobile Developer Specialist:** Provides expertise in mobile development platforms (e.g., iOS, Android) and assists in platform-specific optimizations. ***Reya***

LIFE CYCLE MODEL USED

For this project, we decided to use the Waterfall Model, since the project length is on the shorter side and the requirements are generally well-defined and fixed. The Waterfall Model consists of phases that are executed sequentially. Starting with requirements gathering, the lifecycle then proceeds to the architectural design, followed by the implementation phase, testing phase, and finally the deployment phase. This model provides simplicity and ease of management, as everything is in a chronological order and the milestones are clearly planned out.

RISK ANALYSIS

Every project has its risks, and recognizing them early on is key to staying on track, on budget, and meeting the required quality standards. For this mobile location verification app we're developing we've identified some potential risks, looked at how likely they are to happen, and thought of ways to handle them if they do.

Identified Risks:
1. **Technical Challenges:**
   - **Risk**: Integrating real-time location tracking with identity verification could get tricky. We might face issues like inaccurate location data, slow response times, or challenges in working across both iOS and Android platforms.
   - **Likelihood & Impact:** Medium, High
   - **How to handle it:** We'll start with early prototyping and testing of the most important features—like location tracking and identity verification. This will help us spot problems early. Plus, regular code reviews and using strong, reliable frameworks should keep technical risks under control.
   - **Rationale:** Integrating real-time location tracking with identity verification needs to be accurate and responsive. If location data is off or there are delays, it could impact user trust and overall experience. Also, ensuring it works well on both iOS and Android adds another layer of difficulty. Tackling this risk early through prototypes and testing helps catch and resolve issues before they become bigger problems.
2. **Resource Availability:**
   - **Risk:** The project needs certain resources, like smartphones for testing and the right development environments. Any delays in getting these resources could slow things down.
   - **Likelihood & Impact:** Low, Medium
   - **How to handle it:** We'll make sure we know exactly what resources we need from the start and have them ready early. If there's any delay, we'll have backup options in place to

keep things moving.

- ○ **Rationale:** The development process relies on having the right hardware and software, especially smartphones for testing. If there are delays in getting these resources, it could slow down progress, push back deadlines, and affect testing quality. To avoid this, we need to ensure all necessary resources are available from the start, with backup plans in place to keep things moving smoothly.

3. **Team Communication Issues:**
   - ○ **Risk:** If the team isn't communicating well, it could lead to confusion, duplicated efforts, or missed deadlines.
   - ○ **Likelihood & Impact:** Medium, Medium
   - ○ **How to handle it:** We'll set up clear communication protocols, with regular team meetings, detailed documentation, and tools like Microsoft Teams. Open communication is key and it will be encouraged across the team.
   - ○ **Rationale:** Miscommunication can lead to inefficiencies in the project. If team members aren't clear on their responsibilities or deadlines, it can result in duplicated work or missed deadlines. Establishing clear communication channels and scheduling regular check-ins ensures everyone is aligned and potential misunderstandings are addressed promptly.

4. **Timeline Overruns**
   - ○ **Risks:** Unexpected challenges or delays could cause the project to fall behind schedule.
   - ○ **Likelihood & Impact:** Medium, High
   - ○ **How to handle it:** We'll break the project into manageable phases. Regularly monitoring our progress will allow us to reallocate resources as needed to stay on track. We'll also build buffer time for critical tasks, just in case something takes longer than expected.
   - ○ **Rationale:** Unexpected challenges, such as technical issues or resource delays, can cause timelines to slip. By breaking the project into phases, we can track progress more effectively and identify potential delays early. Adding buffer time to critical tasks ensures that delays don't jeopardize the overall schedule.

5. **Scope Change Risk:**
   - ○ **Risks:** There's a chance that stakeholders might ask for extra features as the project progresses, which could increase our workload and cause delays.
   - ○ **Likelihood & Impact:** Medium, High
   - ○ **How to handle it:** We'll clearly define and document the project scope from day one. If any changes come up, we'll follow a formal process to assess their impact and get approval before making adjustments.
   - ○ **Rationale:** Without a clearly defined scope, there's a risk that stakeholders might request additional features or changes, which can increase workload and cause delays. By clearly defining the project scope and implementing a formal approval process for changes, we can manage expectations and prevent it from overwhelming the team.

6. **Quality Control Issues:**
   - ○ **Risks:** If testing or quality assurance falls short, we could end up delivering a product that doesn't meet the required standards or stakeholder expectations.
   - ○ **Likelihood & Impact:** Medium, High
   - ○ **How to handle it:** We'll implement a thorough testing plan, including unit tests, integration tests, and user acceptance tests. Along the way, we'll have clear quality checkpoints to make sure everything is meeting expectations.
   - ○ **Rationale:** A product that doesn't meet quality standards can damage the project's success. Inadequate testing can lead to bugs, poor performance, or failure to meet stakeholder requirements. Implementing a thorough testing strategy from the start ensures the product meets high standards and reduces the risk of issues after launch.

7. **Stakeholder Misalignment:**
   ○ **Risks:** If the project team and stakeholders aren't on the same page about the objectives or deliverables, it could lead to dissatisfaction or rework.
   ○ **Likelihood & Impact:** Low, High
   ○ **How to handle it:** We'll involve stakeholders in the planning process to make sure everyone is aligned. Regular reviews and updates will help keep us all on the same page throughout the project.
   ○ **Rationale:** If the project team and stakeholders have different understandings of the project's goals or deliverables, it can lead to dissatisfaction, rework, or even failure. Involving stakeholders from the beginning ensures alignment and helps manage expectations throughout the project.

SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS

Hardware Resource Requirements
- Smartphones to download the app and test it
- Laptops with the required IDE's installed
- Internet Connection

Software Resource Requirements

- Operating System:
  ○ Windows 11
- Latest Version of MacOS
- Development Environment:
  ○ Android Studio
  ○ Xcode
- Programming Languages and Frameworks:
  ○ Swift
  ○ JavaScript
  ○ Java/Kotlin
- Web and Server Tools
  ○ Apache
- Version Control:
  ○ Github

DELIVERABLES AND SCHEDULE
1. Research
Activity: Find information related to the project. Discover similar open source applications, discover how iCloud/find works, discover how military C4ISR products perform location functions, and present the results of research to one team
Dependencies: Functional team
Estimated Time: 1 Week
Team Allocation: PM, UI/UX Designer, Frontend Developer, Backend Developer, QA Engineer, Mobile Developer
Deliverables: Research presentation
Rationale: Have a deep understanding and clear plan of the project.

2. Project Initialization

Activity: Define project requirements and objective

Dependencies: Stakeholder meetings to gather requirements

Estimated Time: 2 weeks

Team Allocation: PM, UI/UX Designer, Frontend Developer, Backend Developer, QA Engineer, Mobile Developer

Deliverables: Project requirements document, stakeholder meeting notes, and approvals

Rationale: It is critical to understand project scope, objectives, and constraints to predict the project's progress accurately

3. System Design

Activity: Design Architecture and User Interface

Dependencies: Completed analysis of requirements

Estimated time: 3 weeks

Team Allocation: PM, UI/UX Designer, Frontend Developer, Backend Developer, Mobile Developer

Deliverables: System architecture design, UI mockups, Technical specification document

Rationale: This is the technical foundation for the app and is mandatory for a smooth development process. Scalability, reliability, and easy-use design are prioritized during this phase.

4. Environment Setup

Activity: Configure development environments

Dependencies: Finalized system design

Estimated time: 1 week

Team Allocation: Frontend Developer, Backend Developer, Mobile Developer

Deliverables: Fully configured development and testing environments, Setup guide for developers

Rationale: Developers need a fully functional environment to build and test system

5. Backend Development

Activity: Develop API and Backend Services

Dependencies: Completed system design and environment setup

Estimated time: 6 weeks

Team Allocation: Backend Developer, Mobile Developer

Deliverables: Fully functional backend code, unit test reports, database schema

Rationale: The backend must have robust data handling, user authentication, and location validation services

6. Frontend Development

Activity: Implement mobile app user interface

Dependencies: completed system design and environment setup

Estimated time: 4 weeks

Team allocation: UI/UX Designer, Frontend Developer

Deliverables: Mobile app UI codebase, mobile app navigation

Rationale: Meet mobile design standards and accessibility guidelines

7. Integration and Testing

Activity: Integrate Backend with Frontend, Conduct System Testing

Dependencies: Complete backend and frontend

Estimated time: 4 weeks

Team allocation: UI/UX Designer, Frontend Developer, Backend Developer, QA Engineer, Mobile Developer

Deliverables: Integrated mobile application with backend services, integration testing report

Rationale: Testing ensures both systems work together seamlessly

8. User Acceptance Testing

Activity: Conduct User Acceptance Testing with select users

Dependencies: Completed integration and system testing

Estimated time: 2 weeks

Team allocation: QA Engineer
Deliverables: User Acceptance Testing test plan, User Acceptance Testing feedback report
Rationale: User acceptance testing validates that the system meets the sponsor's requirements and real feedback from the user means the product is reliable.

9. Documentation and Training

Activity: Prepare User Manuals and Conduct Team Training
Dependencies: Completed User Acceptance Testing
Estimated time: 2 weeks
Team allocation: PM
Deliverable: User manuals, training materials, training completion report
Rationale: Ensure users can efficiently use and maintain the system after deployment

10. Deployment

Activity: Deploy the application
Dependencies: Successful User Acceptance Testing and training completion
Estimated time: 1 week
Team allocation: PM, UI/UX Designer, Frontend Developer, Backend Developer, QA Engineer, Mobile Developer
Deliverables: Deployed application, deployment checklist
Rationale: The application is officially released to users

11. Post-Deployment Support

Activity: Provide ongoing support and bug fixes
Dependencies: Deployment
Estimated time: 4 weeks
Team allocation: QA Engineer
Deliverables: Post-deployment support plan, bug fix reports
Rationale: Ensure the system is smooth post-launch and handle all issues in production

MONITORING, REPORTING, AND CONTROLLING MECHANISMS

We will produce three distinct types of reports to ensure proper documentation of our developmental process, those being weekly reports, milestone reports, and lastly our final report.

Weekly Reports
Weekly reports will be utilized as a means of keeping up with individual task distribution, and work from the perspective of us as individuals. We will document each person's developments, roadblocks, and estimated timeline on delivery in order to ensure even balancing of work.

Milestone Reports
Milestone Reports will be attached to each deliverable we submit, which will encompass all of the weekly reports directly feeding into each respective submission.

Final Report
The final report will cover the project's organization, risk analysis, resource requirements, and impact on individuals and organizations, alongside the architecture, design, and test planning. The report also will include evidence of our adherence to engineering standards and configuration management, and concludes with references and acknowledgments, demonstrating our group's application of software engineering principles and learning outcomes throughout our semester.

PROFESSIONAL STANDARDS

Scholastic Dishonesty

Commitment to Original Work
Expected Behavior: Each member is expected to produce original work and contribute their own ideas to the project. When using external resources such as libraries, frameworks, or online content, members must ensure they properly cite sources.

Rationale: Original contributions not only reflect the individual's capabilities but also enhance the overall quality of the project. Proper citation practices acknowledge the efforts of others and uphold the integrity of the team's work. This commitment ensures that the project is a true reflection of the team's skills and knowledge.

Open and Honest Collaboration
Expected Behavior: Team members should engage in honest collaboration, sharing ideas and feedback while providing credit where it is due. If someone utilizes another team member's code or concepts, they must acknowledge that contribution openly.

Rationale: Collaboration is essential in a team environment, and honesty in these interactions builds trust among members. Acknowledging contributions fosters a supportive atmosphere where individuals feel valued and encourages the free exchange of ideas, leading to more innovative solutions.

Addressing Dishonesty
Expected Behavior: If a team member suspects or witnesses any form of academic dishonesty, they are expected to address the issue promptly. This may involve a direct conversation with the individual involved or reporting the concern to ~~a~~ the project sponsor/professor.

Rationale: Allowing dishonesty to go unaddressed can undermine the integrity of the whole project and the team's reputation. Reporting concerns ensures that issues are handled appropriately and reinforces the importance of accountability. It also demonstrates that the team prioritizes ethical behavior and is committed to maintaining high standards.

Meeting Schedule

Punctuality and Preparedness
Expected Behavior: Team members are expected to arrive on time for meetings and come prepared with relevant materials, updates on their tasks, and any questions or issues they wish to discuss.

Rationale: Punctuality shows respect for each other's time and commitments. Being prepared allows meetings to run efficiently and productively, maximizing the time spent together. It also illustrates professionalism and a commitment to the project's success.

Setting Clear Agendas
Expected Behavior: Each meeting should have a clear agenda for all team members to know about prior to the weekly meetings. The agenda should outline the topics to be discussed, the goals for the meeting, upcoming deliverables, and any specific time allocations for each task that needs to be done.

Rationale: A defined agenda helps keep meetings focused and organized, ensuring that all necessary topics are covered within the allotted time. It allows team members to prepare for discussions and contributes to more effective and meaningful conversations.

Action Items and Accountability
Expected Behavior: At the end of each meeting, the team should summarize action items, assigning responsibilities and deadlines for each task. These action items should be documented and shared with all members after the meeting.

Rationale: Clearly defined action items ensure that everyone knows their responsibilities moving forward. This accountability helps maintain momentum between meetings and ensures that tasks are completed in a timely manner, contributing to the overall project goals.

Follow-Up and Progress Check-Ins
Expected Behavior: Team members should provide updates on their assigned action items at the beginning of each meeting, discussing progress, challenges, and any support they may need from the team.

Rationale: Regular follow-ups allow the team to assess progress and identify potential roadblocks early. This transparency fosters collaboration and encourages team members to seek help when needed, ensuring that the project stays on track.

Quality Expectations for Tasks & Deliverables

Time Management and Prioritization
Expected Behavior: Team members are expected to manage their time effectively and prioritize tasks based on project timelines and deliverables. This includes setting realistic deadlines for their work and adhering to the project schedule.

Rationale: Effective time management is essential for meeting project deadlines and ensuring that all tasks are completed to a high standard. Prioritization helps team members focus on critical tasks first, reducing the risk of last-minute rushes that can compromise quality.

Continuous Learning and Improvement
Expected Behavior: Team members should engage in continuous learning by seeking opportunities to learn new skills and knowledge related to software development practices. This can include learning new languages and studying new technologies within the scope of the project to help accomplish deliverables.

Rationale: The technology landscape is constantly evolving, and continuous learning ensures that team members remain up-to-date with best practices and emerging trends. This commitment to learning can lead to higher quality deliverables and more innovative solutions.

Meeting Project Specifications and Deliverables
Expected Behavior: Team members should regularly review the project specifications to ensure their work remains aligned with the given deliverables and that they meet these deliverables on time.

Rationale: Adhering to project specifications ensures that the final product meets the needs of stakeholders and users. Regularly reviewing requirements ensures that all team members are focused on

the same objectives, which is crucial for project success.

Resolving Unacceptable Behavior

Initial Observation and Documentation
Expected Behavior: Team members should remain observant and document any incidents of unacceptable behavior as they occur. This can include noting specific actions, the context in which they occurred, and how they impacted the team or project.

Rationale: Documentation is crucial for understanding the full scope of the issue and provides a factual basis for any discussions or interventions. It helps to avoid misunderstandings and ensures that any actions taken are grounded in specific incidents rather than generalizations.

Private Discussion
Expected Behavior: There should be a private conversation between the involved parties. The team member who observes the behavior should approach the individual in a non-confrontational manner, expressing concerns about the specific behavior rather than making personal judgments.

Rationale: Open dialogue allows for clarification and understanding. The team member may not be aware that their behavior is perceived as unacceptable. A private discussion helps to maintain the dignity of the individual while creating an opportunity for resolution.

Expressing Impact and Concerns
Expected Behavior: During the conversation, the observing team member should clearly articulate how the behavior affects the team dynamic, project progress, and overall morale. It is important to use "I" statements to express personal feelings and observations, such as "I felt concerned when the deadlines were not met because it impacts the entire team."

Rationale: By focusing on the impact of the behavior rather than labeling the person, the conversation stays constructive. This approach encourages reflection and reduces defensiveness, facilitating a more open exchange of ideas.

Collaboratively Identifying Solutions
Expected Behavior: After discussing the issue, both parties should work together to identify practical solutions to prevent future occurrences. This may involve setting clear expectations, adjusting responsibilities, or providing additional support.

Rationale: Collaborative problem-solving empowers team members to take ownership of their behavior and commitments. It fosters a sense of teamwork and demonstrates that the group values each member's contributions.

Involving Team Leadership or Advisors
Expected Behavior: If the behavior persists despite private discussions, or if the situation is severe (e.g., harassment, discrimination, or other serious breaches of conduct), it will be necessary to involve the project sponsor or the professor for mediation.

Rationale: Involving a third party can provide an objective perspective and facilitate a resolution. Those who are in leadership roles often have experience in handling interpersonal conflicts and can offer

guidance on appropriate next steps.

Monitoring Progress
Expected Behavior: The team should monitor the situation over time to ensure that the behavior has improved and that the solutions are effective. Regular feedback sessions can help maintain open lines of communication.

Rationale: Continuous monitoring demonstrates a commitment to resolving the issue and reinforces the importance of maintaining a positive team environment. It allows for adjustments to be made if problems recur.

EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT
- We will be using GitHub to keep a track of who is working on what part. It is a configuration management tool that we will use. Also, it will ask for approval from 2 group members before checking in a change to the document.
- https://github.com/AlvinsGitHubb/Location-Verification-App

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS
- Project Management Plan:
  - IEEE Std 1058-1998 for Software Project Management Plans
  - IEEE Std 830-1998 for Software Requirements
  - IEEE Std 1471-2000 for Software Architecture
  - IEEE Std 1016-1998 for Software Design
  - IEEE Std 829-1983 for Software Testing
  - PMBOK® Guide: Project Management Body of Knowledge
  - IEEE Std 12207: Software Life Cycle Processes
  - IEEE Std 15939: Measurement Process
  - ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
    - Life Cycle Processes
    - Requirements Engineering
- ~~Requirements Documentation~~
  - ~~IEEE Std 830-1998: Software Requirements~~
  - ~~IEEE Std 29148: Requirements Engineering~~
  - ~~ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering~~
    - ~~Life Cycle Processes~~
    - ~~Requirements Engineering~~
- ~~Architecture Documentation~~
  - ~~IEEE Std 1471-2000: Software Architecture~~
  - ~~ISO/IEC/IEEE Std 42030:2019: Software, Systems and Enterprise~~
    - ~~Architecture Evaluation Framework~~
- ~~Detailed Design Documentation~~
  - ~~IEEE Std 1016-1998 (Revision 2009): Software Design~~
- ~~Test Plan~~
  - ~~IEEE Std 829-1983: Software Testing~~

- ~~ISO/IEC/IEEE Std 29119-1 (Revision 2022): Part 1 - Software Testing General Concepts~~
- ~~ISO/IEC/IEEE Std 29119-2 (Revision 2021): Part 2 - Test Process~~
- ~~ISO/IEC/IEEE Std 29119-3 (Revision 2021): Part 3 - Test Documentation~~
- ~~ISO/IEC/IEEE Std 29119-4 (Revision 2021): Part 4 - Test Techniques~~
- ~~Final Project Report~~
  - ~~All the above without duplicate~~
  - ~~All source on GitHub~~
  - ~~All documentation on project Google folder~~
  - ~~Include test results on project Google Folder~~

ADDITIONAL REFERENCES

~~https://www.getapp.com/all-software/address-verification/~~
~~https://www.sciencedirect.com/science/article/pii/S0148296321008298~~
Belligoni, Sara, et al. "Privacy and Security Concerns with Passively Collected Location Data for Digital Contact Tracing among U.S. College Students." *PloS One*, U.S. National Library of Medicine, 22 Nov. 2023, www.ncbi.nlm.nih.gov/pmc/articles/PMC10664924/.
Priego, Nuria Rodríguez, et al. "Sharing but Caring: Location Based Mobile Applications (LBMA) and Privacy Protection Motivation." *Journal of Business Research*, Elsevier, 23 Nov. 2021, www.sciencedirect.com/science/article/pii/S0148296321008298.
Wisniewski, Pamela, et al. "Predicting Smartphone Location-Sharing Decisions through Self-Reflection on Past Privacy Behavior." *OUP Academic*, Oxford University Press, 22 Sept. 2020, academic.oup.com/cybersecurity/article/6/1/tyaa014/5910009.
Yan, Shihao, and Robert Malaney. "Location Verification Systems in Emerging Wireless Networks." *arXiv.Org*, 12 July 2013, arxiv.org/abs/1307.3348.

- Project Management Plan
  - Larson, E. and Gray, C., 2014. Project Management: The Managerial Process. McGraw Hill
  - o Humphrey, W.S. and Thomas, W.R., 2010. Reflections on Management: How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself. Pearson Education
- ~~Requirements Documentation~~
  - ~~Lamsweerde, A.V., 2009. Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley~~
- ~~Architecture Documentation~~
  - ~~Lattanze, A.J., 2008. CRC Press~~
  - ~~Bass, L., Clements, P. and Kazman, R., 2003. Software Architecture in Practice. Addison-Wesley~~
- ~~Detailed Design Documentation~~
  - ~~Larman, C., 2012. Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development. Pearson Education~~
  - ~~Hyman, B., 1998. Fundamentals of Engineering Design. New Jersey: Prentice Hall~~

- ○ Simon, H.A., 2014. A Student's Introduction to Engineering Design: Pergamon Unified Engineering Series (Vol. 21). Elsevier
- Test Plan
  - ○ Jorgensen, P.C., 2013. Software Testing: A Craftsman's Approach. Auerbach Publications
  - ○ Mathur, A.P., 2013. Foundations of Software Testing, 2/e. Pearson Education
- Final Project Report
  - ○ All the above without duplicate.

Appendix A.

The following provides a professional standards guideline for the teams. This guideline may be tailored. The professional standards must be agreed upon by each member in the team.

Guideline:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the *option* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as "too busy" or "I forgot", or "my dog ate my design model" are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)