Group 2 Project Plan

SWE 6623: Software Engineering

Paul Hudlow, Ilyas Kure, Durrell Lyons, Joshua Nunez

February 28, 2016

1 Introduction

1.1 Overview

The high-level task is to build a web application that serves as a central company calendar to help employees keep track of important events. The project, including requirements analysis, system design, program design, implementation, and testing will be carried out by a four-person team starting on February 29th, 2016 and finishing with final deliverables on April 17th, 2016.

1.2 Project Deliverables

The major milestones for this project are the completion of Requirements, Design, and Testing. The deliverables for each milestone are shown in the following table.

Completed Milestone	Deliverable	Due Date
Requirements	Requirements Specification	3/13/2016
Program Design	Design Document	3/27/2016
Implementation & Testing	Program Code	4/17/2016
	Unit Test Code	
	Test Plans	

More information about documentation deliverables is given in section 4.2.

1.3 Evolution

After the submission of the project plan any changes must be reviewed and discussed by the entire team. The team leader will ultimately approve changes unless there is a significant dispute which would be addressed by Dr. Pournaghshband.

1.4 Reference Material

The IEEE Project Management Plan outline was used as a structure for the plan documentation. Lecture notes from Dr. Hassan Pournaghshband and several chapters of *Software Engineering* by S.L. Pfleeger and J. M. Atlee provided guidance for project planning activities.

1.5 Definitions and Acronyms

Acronym/Initialism	Meaning
VM	Virtual Machine

2 Project Organization

2.1 Process Model

The deliverable schedule imposed by the customer is relatively linear in nature. For this reason the process will closely resemble the Waterfall model. However, since the team members will be consistent throughout the project, each will be expected to contribute during every phase of the process. Also, a prototype will be built during the design phase of the project as a proof of concept. Unlike typical prototypes built for the purpose of refining requirements, the purpose of this prototype will be to validate the practicality of the system design.

2.2 Organizational Structure

Given the small size of the project team, the structure is very simple. Paul Hudlow will assume the role of team leader. Other team members will not have formally assigned roles, but rather collaborate dynamically depending on changing project needs.

2.3 Organizational Boundaries and Interfaces

Team members will communicate regularly using online discussion boards and meet at least once per week through using Internet telephony conferencing. Project files will be stored in a shared online repository accessible to all team members. The team will work autonomously with only limited guidance from Dr. Pournaghshband.

2.4 Project Responsibilities

The team leader will be responsible for collecting, formatting, and delivering project documentation. He will also be responsible for managing and assigning tasks as necessary. Other responsibilities will be flexible depending on project needs and individual skills.

3 Managerial Process

3.1 Management Objectives and Priorities

The main management objective is to obtain as high a grade as possible for the project as a whole. Thorough and high-quality documentation is one of the highest priorities since the majority of the overall grade will be determined by the delivered documentation. Meeting deadlines for major project deliverables is a priority since the schedule cannot slip and missing deadlines will result in an automatic deduction from the earned grade.

3.2 Assumptions, Dependencies, Constraints

Some team members are largely unfamiliar with the chosen languages and environments used for the project. This makes it critical for tasks to be flexibly managed to allow each team member to contribute using his own skillset.

Team members are physically remote, and are all committed to different schedules regarding project involvement. Thus centralized online coordination tools will be critical for project success.

3.3 Risk Management

Because of the small size of the project and the short schedule, unexpected absences of team members are a significant risk. In order to mitigate this risk team members will be encouraged to

check code into the central repository frequently. In addition, task completion will be tracked online so that all team members will have some awareness of progress.

Another risk to meeting project deadlines are technical roadblocks due to limited experience of team members. A proof of concept prototype will be constructed in the design phase of the project to help mitigate this risk. Team members will also be encouraged to post any resources used during development to keep other members appraised of technical approaches.

3.4 Monitoring and Control Mechanisms

Project tasks and progress will be monitored through an online issue tracking system. The entire team will meet once per week through Internet telephony conferencing. Other communication will be conducted through email and an online discussion board.

3.5 Staffing Plan

The project will be completed using an autonomous team made up for the following four members. Paul Hudlow, Ilyas Kure, Durrell Lyons, Joshua Nunez.

4 Technical Process

4.1 Methods, Tools, Techniques

4.1.1 Hardware

Development of the system will take place on standard desktop grade computers. No special hardware is needed for development or testing of the system as the system will be a standard web based application.

4.1.2 Languages

The languages used to develop the system will include PHP, SQL, HTML, CSS, and Javascript.

4.1.3 Software

A development virtual machine will be built with Linux for the prototype to be assembled during the design phase of the project. The virtual machine will include all tools necessary to develop, built, and run the project. This VM will be distributed to team members for individual customization and use.

A PHP edition of Eclipse (https://eclipse.org/pdt/) will be used to organize the project, a MySQL database will be set up to store the application data, and Apache will supply the necessary web services.

4.1.4 Project Coordination

A private GitHub repository will be used to store the application code. Each developer will maintain a Git workspace on his own copy of the development VM. Each major change should be submitted to a separate branch and then reviewed and approved by at least one other team member before being merged to the main line code. Special exceptions to this rule may be approved by the team leader.

Debugle.com will be used to coordinate project tasks. The team leader will be responsible for creating, managing, and assigning tasks as necessary. Every team member will be responsible for

reviewing task comments before working and posting any comments that are relevant to the rest of the team.

4.1.5 Team Structure

The team will consist of one team leader and several team members working under the leader's direction. The team leader will be responsible for assigning tasks to the team members as well as contributing to the end product. The team leader will insure the team delivers on its promised by the dates specified in the Project Plan.

4.2 Software Documentation

The following sections outline the documentation that will be completed at each milestone as shown in section 1.2. There are four members on this team and each member will review all documentation produced before it is submitted.

4.2.1 Requirements Specification

The requirements specification document will provide a detailed list of all project requirements categorized and numbered for tracing through project completion and testing.

4.2.2 Design Document

The design document will include an explanation of the overall design of the system as well as detailed design of the user interface, the data access, and the business logic functionality.

4.2.3 Test Plan

The test plan document will outline every test scenario chosen and specify which requirements are tested by each. It will be written to clearly show that all requirements documented in the specification are tested at least once during testing. This document will also briefly outline the unit tests that were developed alongside the code.

4.3 Project Support Functions

The team will be autonomous and self-supporting. Testing and quality assurance tasks will be managed in the same way as development tasks.

5 Schedule and Budget

5.1 Work Packages

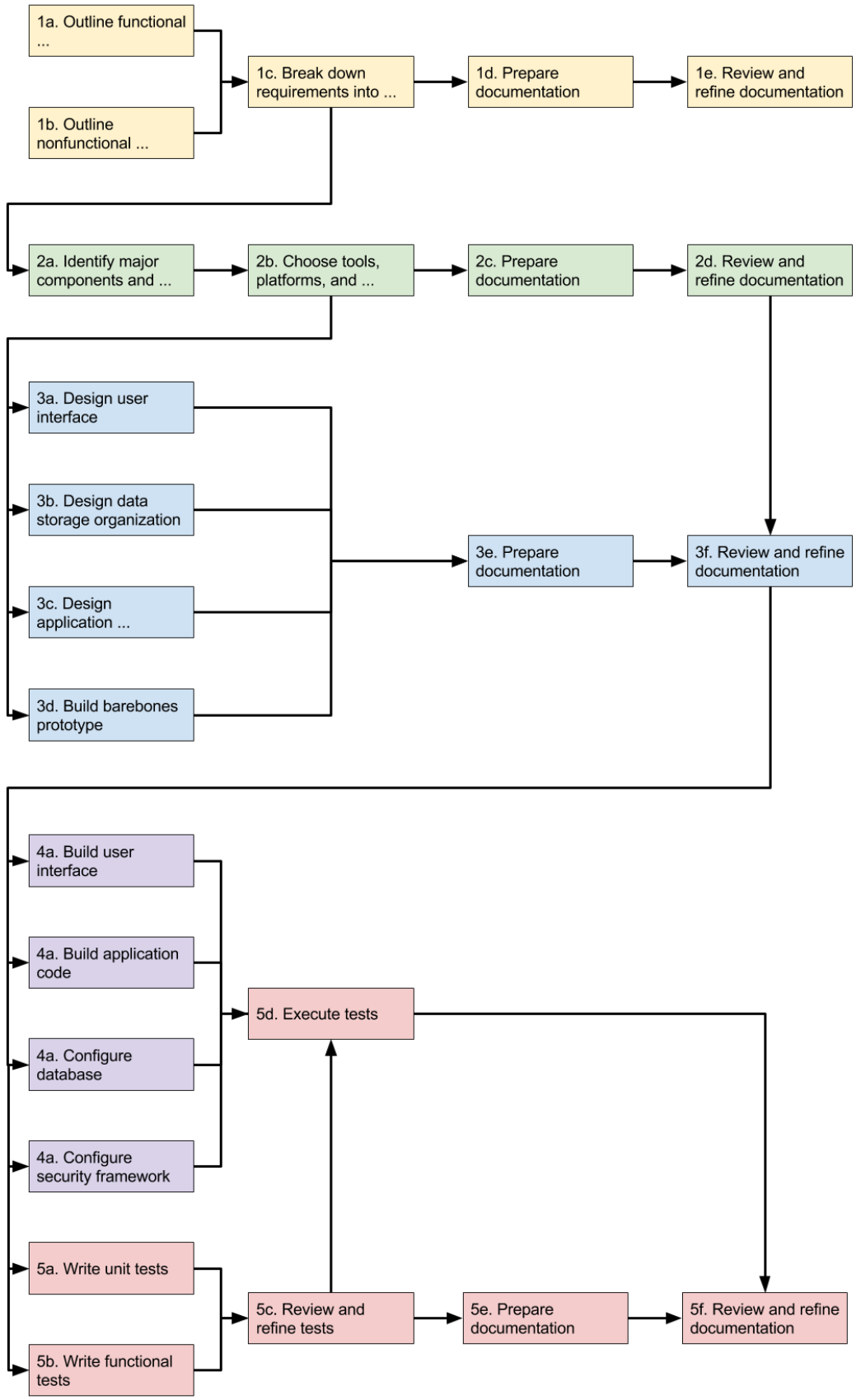
The following list shows the major work packages required for project completion.

1. Requirements Analysis
    a. Outline functional requirements
    b. Outline non-functional requirements
    c. Break down requirements into specifics
    d. Prepare documentation
    e. Review and refine documentation
2. System Design
    a. Identify major components and interactions
    b. Choose tools, platforms, and frameworks
    c. Prepare documentation

    d. Review and refine documentation
3. Program Design
    a. Design user interface
    b. Design data storage organization
    c. Design application organization
    d. Build barebones system prototype
    e. Prepare documentation
    f. Review and refine documentation
4. Program Implementation
    a. Build user interface
    b. Build application code
    c. Build database tables
    d. Configure security framework
5. Testing
    a. Write unit tests
    b. Write test plans
    c. Review and refine tests
    d. Execute tests
    e. Prepare documentation
    f. Review and refine documentation

5.2 Dependencies

The project dependencies are defined by the following chart which includes every work package listed in the previous section.


1a. Outline functional ...
1b. Outline nonfunctional ...
1c. Break down requirements into ...
1d. Prepare documentation
1e. Review and refine documentation
2a. Identify major components and ...
2b. Choose tools, platforms, and ...
2c. Prepare documentation
2d. Review and refine documentation
3a. Design user interface
3b. Design data storage organization
3c. Design application ...
3d. Build barebones prototype
3e. Prepare documentation
3f. Review and refine documentation
4a. Build user interface
4a. Build application code
4a. Configure database
4a. Configure security framework
5d. Execute tests
5a. Write unit tests
5b. Write functional tests
5c. Review and refine tests
5e. Prepare documentation
5f. Review and refine documentation

5.3 Schedule

The following table shows the schedule of completion for each work package. Note the deliverable due dates are shown in bold font.

Project Phase	Work Package	Due Date
Requirements Analysis	Outline functional requirements	3/4/2016
	Outline non-functional requirements	3/4/2016
	Break down requirements into specifics	3/10/2016
	Prepare documentation	3/12/2016
	Review and refine documentation	**3/13/2016**
System Design	Identify major components and interactions	3/14/2016
	Choose tools, platforms, and frameworks	3/17/2016
	Prepare documentation	3/19/2016
	Review and refine documentation	3/20/2016
Program Design	Design user interface	3/25/2016
	Design data storage organization	
	Design application organization	
	Build barebones system prototype	3/26/2016
	Prepare documentation	
	Review and refine documentation	**3/27/2016**
Program Implementation	Build user interface	4/11/2016
	Build application code	
	Build database tables	
	Configure security framework	
Testing	Write unit tests	4/11/2016
	Write test plans	
	Review and refine tests	
	Execute tests	
	Prepare documentation	4/15/2016
	Review and refine documentation	**4/17/2016**