HR Employee Tracking Application

Project Plan

Revision History

Name	Role	Purpose	Date	Document Status
Zach Comstock	Documentation Engineer	Initial Creation	9/3/17	Draft
Zach Comstock	Documentation Engineer	Revised formatting	10/30/17	Draft
Zach Comstock	Documentation Engineer	Added Project Plan, WBS, and Milestones	11/1/17	Draft
Zach Comstock	Documentation Engineer	Refined comprehensive plan	11/14/17	Complete

Table of Contents

Project Scope

The ultimate goal of this project, as it is understood, is to create a software application that is able to track and monitor an employee's current standing in the company. The project shall hereby be referred to as the "HR Employee Tracking Application". The target user group for the HR Employee Tracking Application is expected to primary be Human Resources. However, the application will also be used by managers and general employees, with restricted functionality of course. The HR Employee Tracking Application is intended to assist Human Resources with the management and tracking of employees' and managers' productivity and employment information vacation and sick hours. Human Resources shall also be able to use the HR Employee Tracking Application to approve or employees' and managers' requests, such as vacation, raise, or promotion requests.

The application is meant to be a web based application containing a frontend to handle the views, a backend to process the logic, and a database to store the data. The database should handle modifications to the information (add, read, edit, delete) stored within based on the permissions granted to each specific user.

In order to create a fully functioning HR Employee Tracking Application, the project must include the following features per each user group:

User Group	Functionality Needs
Employees	Employees should be able to view all of this information and be able to file a discrepancy complaint if something is inconsistent with what they know. HR Should be able to send this complaint to the manager for them to review. Employees should also be able to request vacation and it should show that the vacation days are subtracted from the amount they accumulated over the course of their work. In the same way, employees should be able to request sick leave. The manager should get a notification that their employee is requesting sick leave and should be able to easily see whether or not the request can be confirmed or denied. Employees should be able to also quit their job whenever they so choose by putting in a two weeks notice.
Managers	The managers should be able to write reviews on their employees, approve or deny sick leave & vacations, and see whether or not an employee is in good standing with the company. The manager should also be able to give raises to an employee, which should be cleared by HR before being set into place.
Human Resources	The HR Department should be able to terminate employees, but still keep their records on file. They should be able to approve of vacations that exceed the number set by the company and a negative result should go into the manager's file if that occurs. HR should be able to send a manager a warning about potentially failing employees. HR should also be able to accept or deny increased in salary and decreases in salary.

This project shall provide a convenience for Employees and Human Resources alike, that we believe will be used heavily. By allowing Human Resources to manage and monitor employees with ease, we hope to reduce

the cost and total expended manhours, which should in turn free up more resources. The reduced costs will help contribute to profits and the reduced stress on the employees will allow them to attend to other duties and improve their effectiveness.

The rest of this document contains a rough estimate of the total overall cost of the Software Project Tracker from beginning to end. All costs are measured in man-hours and based on previous project experience. Please keep in mind that this is only an estimate and these predictions are subject to change.

Project Deliverables

The project will be separated into 8 critical deliverables as defined in the project milestones on page x of this document. Each deliverable has a defined amount of effort and resources that will be needed for its' completion. The following table will define each major deliverable, identify the necessary tasks leading up to this deliverables' competition, and declare the amount of time and effort needed to complete each deliverable. All deliverables are subject to revision after said deliverable's due date.

Deliverable Name	Description	Tasks	Effort expended	Due by
Software Requirements Specification	The project requirements have been defined, reviewed, revised, and finalized.	1. Gather User stories 2. Test User stories 3. Revise User stories 4. Define Requirements 5. Evaluate Requirements 6. Finalize SRS	13 estimated man-hours in total	10-9-2017
UML Use Case Diagram	High level design of the system has been completed and is easily viewable through the UML Use Case Diagram	1. Compose list of all major features of the system 2. Create UML Use Case diagram to ensure all features connect and are properly defined.	7 estimated man-hours in total	10-16-2017

Use Cases	Each major feature of the application will have defined use cases that will further validate the UML Use Case Diagram and finalize the system design.	1. Create a use case document for each feature in the use case diagram. 2. Test that the use cases all work with the current design. 3. Correct any logical errors.	6 estimated man-hours in total	10-23-2017
Software Design Document	The entire system design will have been finished at this point. (use cases and use case diagram). Data flow diagrams and user interface designed will also be refined and completed at this point.	1. Create a data flow diagram for each use case. 2. Wireframes and storyboard created for each use case.	15 estimated man-hours in total	10-30-2017
Prototype	A working prototype of the frontend of the application will be available for stakeholders to demo; this will be the stakeholders' last chance to redefine or add new requirements.	1. Decide on programming language and framework for development. 2. Set up development environment for each team member 3. Assign roles 4. Design database 5. Develop critical prototype views for stakeholders to demo	11.5 estimated man-hours in total	11-7-2017
Finished Application	At this point the projects' front-end and back-end will be complete. Most if not all requirements will be complete.	1. Create database 2. Create back-end 3. Connect back-end to front-end	10 estimated man-hours in total	11-20-2017

Testing Document	A document detailing all the tests that need to be completed for the application.	1. Identify all expected input/output for each user case. 2. Document each test case.	10 estimated man-hours in total	11-24-2017
Issues/Errors Documentation	All tests will have been completed and any issues or errors will be documented.	1. Run through all tests for the system. 2. Document any issues or errors	15 estimated man-hours in total	11-27-2017
User Manual	A user manual /guide detailing how a general user can operate the application with be finished.	1. Identify each user and all its' use cases. 2. Document each use case in plain English	10 estimated man-hours in total	12-1-2017
Tested and Finished Application	All issues or error document will be resolved at this point in time	1. Correct all issues or errors that have been documented	5 estimated man-hours in total	12-3-2017

Project Plan

In order to ensure that the project is delivered on time, we will develop using a traditional waterfall model. Therefore, it is critical for the client to assist in the creation of the Requirements Specification and ensure that all business needs are represented correctly. Once signed-off on, the client will later have a second chance, after the completion of the application prototype, to alter or add new requirements (at a greater expense). At this point in time, all requirements will be set in stone and the application will be created in accordance to the original specification.

A maximum of 15 hours a week will be allocated to working on the project, depending on the progress of the software project. At this point in time we do not plan to exceed any more than 15 manhours a week; the estimated man hours are listed above. However, we anticipate that some issues may arise which will require more than the estimated manhours. Therefore, in some instances the manhours will be hirer than expected, but not exceeding 15 hours a week; this worst-case scenario will be represented in the cost estimated at the end of this document.

Work Breakdown Structure

Phase 1: Requirements Gathering / Analysis	Phase 2: Design	Phase 3: Development	Phase 4: Testing
Step 1.1.0: Iterate on the Project Vision to conform more closely to the stakeholders' needs. (Cost 2)	Step 2.1.0: Compose list of all objects that will be involved in the system (Cost 3)	Step 3.1.0: Create general user section / view (Cost 5)	Step 4.1.0: Create Testing Document (Cost 10)
Step 1.1.1: Turn project summary into clearly defined user-stories (Cost 2)	Step 2.2.0: Create UML Use Case Diagram (Cost 4)	Step 3.1.2: Create manager section / view (Cost 5)	Step 4.2.0: Run through all use cases (Cost 9)
Step 1.2.0: Test user-stories against initial Project Description and stakeholder's expectations. (Cost 1)	Step 2.3.0: Compose written use cases for key features (Cost 4)	3.1.3: Create Human Resources section / view (Cost 5)	Step 4.2.1: Verify that all use cases work (Cost 3)
Step 1.3.0: Revise user-stories and project vision as needed (Cost 1-3)	2.4.0: Re-evaluate UML Activity Diagram for full coverage (Cost 2)	3.2.0: Create Database for information storage (Cost 4)	Step 4.2.2: Document any errors or issues found (Cost 3)
Step: 1.4.0: Verify that the revised user-stories are correct (Cost 1)	2.5.0: Decide on what utilities and coding languages will be best suited for the system's needs (Cost 2)	3.3.0: Create back-end for communicating with the database (Cost 3)	Step 4.3.0: Fix any errors and retest (Cost 5)
Step 1.5.0: Compose the Software Requirements Specification (Cost 4)	2.6.0: Set up identical development environment for all team members (Cost 2)	3.4.0: Connect each separate module together (Cost 3)	Step 4.4.0: Create User Manual (Cost 10)
	2.7.0: Assign roles for each team member (Cost 1)		
	2.8.0: Design database structure (Cost 2)		
	2.8.1: Confirm that current system design will be able to handle all use cases (Cost 2)		
	2.9.0: Design dataflow diagrams for each use-case (Cost 5)		

	2.9.1: Create wireframes / user stories for each use-case (Cost 5)		
	2.10.0: Prototype each section / view (Cost 2.5)		

Milestones

1.5 Software Requirements Specification is composed Software Requirement Specification	October 9th
2.2 UML Use Case Diagram is finished 4)	October 16th
2.3 Written use cases for key features in the system have been written written use cases for key features (Cost	October 23rd
2.9.1: Software Design Document is finished	October 30th
2.10 Prototype is finished	November 6th
3.1 Basic features of the application are accessible for early showcasing to stakeholders	November 13th
3.4 Application is created and ready for testing	November 20th
4.1 Application functionality test and errors are documented	November 24th
4.3 Errors have been correct and the application is ready for delivery	November 27th
4.4 User manual is completed	December 1st

According to the ES, EF, LS, LF Table below, the earliest estimated delivery cost is 110.5 and the latest estimated delivery cost is 112.5 (subject to change). Assuming the project is officially started on October 3rd 2017 and set to be finished by December 3rd 2017, our team will have roughly 9 weeks to complete the HR Employee Tracking Application. Therefore, provided that a minimum of 10 man hours are allocated to this project each week, the project will be delivered on time.

If this practice is adopted, then the team will hit a new milestone every 5-7 days, which means that the stakeholders will be able to demo a prototype of the Software Project Tracker as early as 5 weeks into development (the first week of November).

ES, EF, LS, LF Table

Activity	Earliest Start	Earliest Finish	Latest Start	Latest Finish
Iterate on the Project Vision to conform more closely to the stakeholders' needs.	0	2	0	2
Turn project summary into clearly defined user-stories.	2	4	2	4
Test user-stories against initial Project Description and stakeholder's expectations.	4	5	4	5
Revise user-stories and project vision as needed	5	6	5	8
Verify that the revised user-stories are correct	6	7	8	9
Compose the Software Requirements Specification	7	11	9	13
Compose list of all objects that will be involved in the system	11	14	13	16
Create UML Use Case Diagram	14	18	16	20
Compose written use cases for key features	18	22	20	24
Re-evaluate UML Activity Diagram for full coverage	22	24	24	26
Decide on what utilities and coding languages will be best suited for the system's needs	24	26	26	28
Set up identical development environment for all team members	26	28	28	30
Assign roles for each team member	28	29	30	31
Design database structure	29	31	31	33
Confirm that current system design will be able to handle all use cases	31	33	33	35
Design dataflow diagrams for each use-case	33	38	35	40
Create wireframes / user stories for each use-case	38	43	40	45
Prototype each section / view	43	45.5	45	47.5
Create general user section / view	45.5	50.5	47.5	52.5
Create manager section / view	45.5	50.5	47.5	52.5
Create Human Resources section / view	45.5	50.5	47.5	52.5
Create Database for information storage	45.5	49.5	47.5	51.5
Create back-end for communicating with the database	49.5	52.5	51.5	54.5
Connect each separate module together	52.5	55.5	54.5	57.5
Create Testing Document	55.5	65.5	57.5	67.5
Run through all use cases	65.5	74.5	67.5	76.5
Verify that all use cases work	74.5	77.5	76.5	79.5
Document any errors or issues found	77.5	80.5	79.5	82.5
Fix any errors and retest	80.5	85.5	82.5	87.5
Create User Manual	85.5	95.5	87.5	97.5

Team Roles

Name	Roles
Zach Comstock	Documentation/Project Manager
Eddy Jean	Back-End Engineer
Ruth Petit-Bois	Front-End Engineer / Design and Testing Engineer
Benjamin Kaguwo	Database Engineer / Testing Engineer
Lisian Ajroni	Back-End Engineer

Detailed Schedule and Cost Estimate

A detailed schedule of the project is displayed below. We have already prepared additional resources in the unlikely situation that tasks take more man-hours than previously estimated. Therefore, we do not anticipate any critical variations in the schedule.

Week Number	Task(s)	Assigned To	Est. Cost	Max Cost	Finished by
Week 1	1. Define user-stories 2. Test user-stories 3. Revise user-stories 4. Finalize user-stories 5. Compose SRS document 6. Make design prototype	Zach Comstock, Ruth Petit-Bois	11-13	15	10/9/2017
Week 2	1. Compose list of system objects 2. Create UML Use Case Diagram	Zach Comstock	7	15	10/16/2017
Week 3	1. Compose written use-cases for key features 2. Revaluate UML use-case diagram	Zach Comstock	6	15	10/23/2017
Week 4	1. Setup development environment 2. Design Database 3. Test system design 4. Create prototype	Lisian Ajroni, Ruth Petit-Bois, Benjamin Kaguwo	12.5	15	10/30/2017
Week 5	1. Create	Ruth Petit-Bois	15	15	11/6/2017

	general user section / view 2. Create manager section / view 3. Create Human Resources section / view				
Week 6	1. Create Database for information storage 2. Create back-end for communicating with the database 3. Connect each separate module together	Eddy Jean, Ruth Petit-Bois, Benjamin Kaguwo	9	15	11/13/2017
Week 7	1. Create Testing Document	Benjamin Kaguwo	10	15	11/20/2017
Week 8	1. Run through all use cases 2. Document any errors or issues found	Zach Comstock	10	15	11/24/2017
Week 8 pt. 2	1. Create user-manual	Benjamin Kaguwo	10	15	11/27/2017
		Cost Estimates	90.5 – 92.5	135	