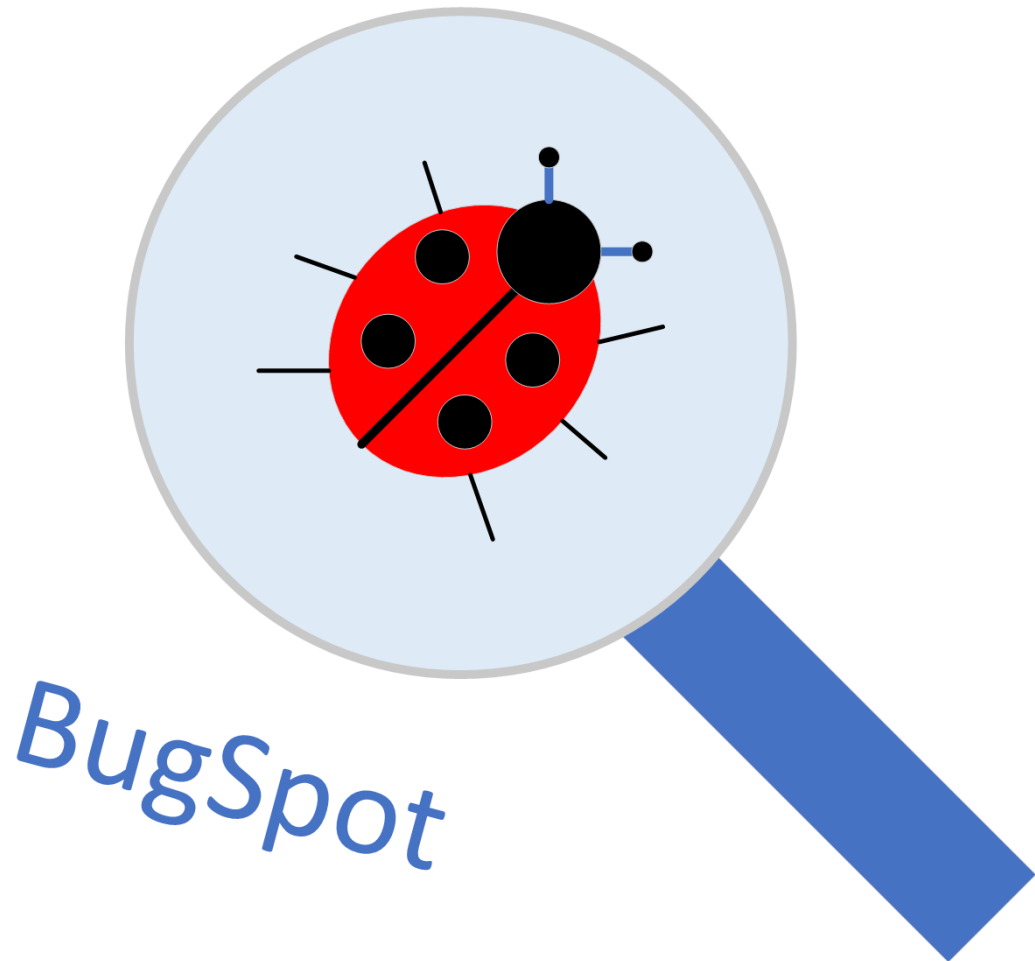


Software Development Plan Bug Report System



By USN Software AS

Table of Contents

1. Introduction	3
2. Product Description	4
3. Project Organization	5
4. Coding Conventions	6
5. Risk Analysis	7
6. Work Breakdown	8
7. Project Schedule	9
8. Monitoring and Reporting Mechanisms	10
9. Tools	11

1. Introduction

Our developers at USN Software are working on a new system for tracking bugs and features, called BugSpot. This system is targeted for developers needing feedback from their customers to make improvements to the product or service they are providing. In addition to users being able to report and send in bugs to the developers of a product or service, the developers should also be able to post and show their users new and exciting features they have been working on.

This document presents the organization, required resources, risk assessment, work breakdown and estimated scheduling for the project.

2. Product Description

The product is targeted as an application for tracking and management of bugs and features for software developers. **Error! Reference source not found.** shows a sketch of planned system.

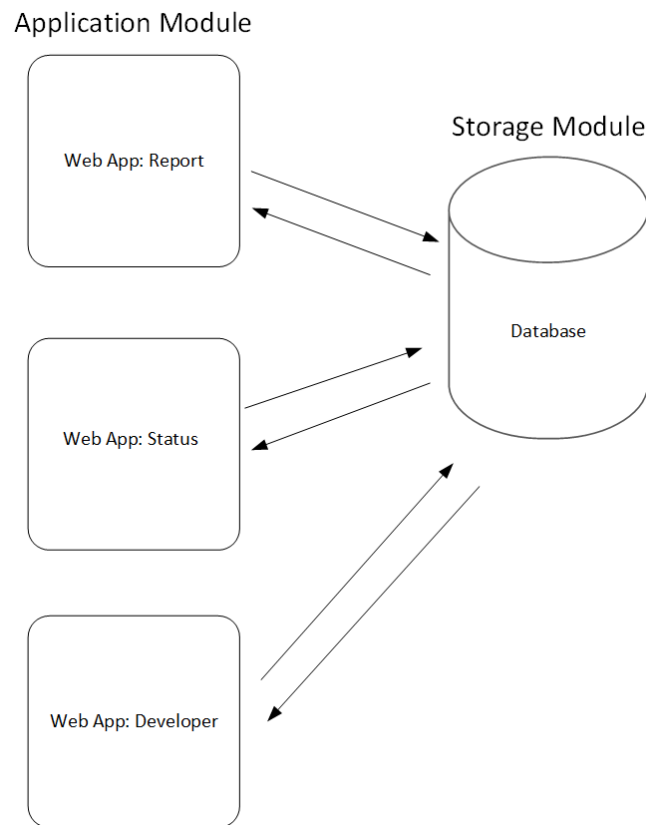


Figure 2-1: General overview of product

The BugSpot app is an application where users can submit reports about bug(s) they encounter. These reports are then received by the developers where they can review whether the reports are something they want to investigate or not. The system itself will be very simple to use and understand from a user perspective as the form they are submitting will be very intuitive to use.

This bug tracker system will also have a “Features” section or page, where the developers can post whenever new functions, implementations and bug fixes are added. In the posts they will also be able to add comments/descriptions if they need to explain further in depth on certain aspects of a new feature. Additionally, users will be able to submit a request for a particular feature from the developers.

All this information will be stored in a storage module like a database for ease of access.

3. Project Organization

To produce an application like this bug tracker, we need people with skills within the following fields:

- Programming (C#)
- Web development (HTML)
- Database development (SQL)

Our team consists of three people developing the bug tracker application, they are experienced within the fields required to develop the application. More information about them below or read upon their resume on Microsoft Teams.

Team members:

[Justinas Lapenas – Project Leader](#)

Experience: C#, Python, SQL & HTML programming.

[Ole Magnus Woje Jensen – Project Assistant Leader](#)

Experience: C#, SQL, HTML, .NET, Windows Server, Web Hosting

[Marius Eriksen – Lead Programmer](#)

Experience: C#, Python, SQL, HTML, Azure.

4. Coding Conventions

To keep the coding as uniform as possible between all project group members certain standards are established for coding of various modules. The list bellow showcases the basic coding conventions for the project.

- C#
 - Variable naming uses camelCase
 - Method naming uses PascalCase
 - Methods must use simple descriptive names
 - Methods should have commentary which describes what the code bellow does
 - Class name, parameters and methods uses PascalCase
- SQL (Database)
 - SQL commands uses all capital letters (INSERT, DELETE, CREATE, etc.)
 - Stored Procedures name must include the usp abbreviation before PascalCase name (uspProcedureName)
 - View names must use PascalCase
 - In case of multiple (more than 2) tables included in a view the view name must be descriptive of what information the view shows (CUSTOMER, POSTALADDRESS & POSTALNUMBER would be named "CustomerInfo")
 - Database table names must use all capital letters
 - Database table column names must use PascalCase

5. Risk Analysis

Risks may threaten the project, the software that is being developed or the organization. Even losing a team member can be a business risk because an experienced engineer's reputation may be a critical factor in winning new contracts. In general, high severity risks should always be considered, as should all serious risks that have more than a likely probability of occurrence. Table 5-1 shows estimated probability and severity of possible risks. Table 5-2 is the matrix tool used to identify each of the risks and their factors regarding probability and severity.

Table 5-1: Risk Analysis

	Risk	Criticality	Severity	Probability
	<i>Hacker attack</i>	100%	High	Very Likely
	<i>Loss of data</i>	100%	High	Very Likely
	<i>DoS attack</i>	80%	High	Likely
	<i>System will not be finished within the deadline</i>	80%	High	Likely
	<i>Unforeseen absence</i>	33%	Medium	Likely
	<i>Miscommunication/misunderstanding</i>	33%	Medium	Likely
	<i>Tool malfunction</i>	10%	Low	Unlikely
	<i>Technical challenges</i>	10%	Low	Likely
	<i>User login problems</i>	10%	Low	Unlikely

Table 5-2: Risk Matrix

	Low	Medium	High
<i>Very Likely</i>	0	0	2
<i>Likely</i>	1	2	2
<i>Unlikely</i>	2	0	0

6. Work Breakdown

The work required to complete the project is broken down into the following:

- Project preparation/kick-off
 - Group Introduction
 - Role Assignments
 - Software installation
- Planning/discussion
 - Software Development Plan
 - Brainstorming
 - Risk analysis
 - Gantt chart
 - Setting standards/formatting for documentation & coding
- Database design
 - Database structure Model
 - Database prototype
- Application design
 - Interface
 - API to database
 - Application prototype
- Testing
 - Functionality testing
 - Bug testing
- Customer/User Feedback
 - Prototype presentation
- Maintenance
 - Bug fixes
 - Software updates

7. Project Schedule

The following Gantt chart, shown in **Error! Reference source not found.** is an estimate of project duration and shows when various parts of the project should be completed. During the project some changes may occur. Changes to the schedule will be noted further bellow by date of occurrence.

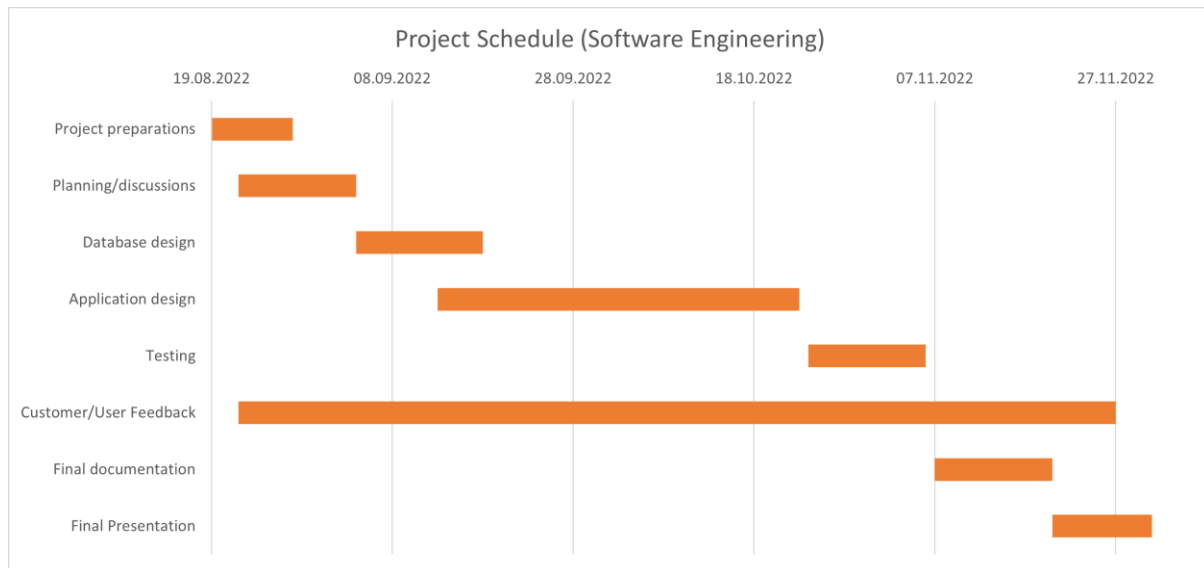


Figure 7-1: Gantt chart showing the project schedule

Note: The end user feedback is taken into consideration from the planning phase.

8. Monitoring and Reporting Mechanisms

To ensure that the project system follows the planned route of development various tools will be used to monitor the group for deviations from the original plan. For this project Azure Dev Ops will be utilized extensively to keep track of the source code, assign tasks-at-hand and development of the project. Figure 8-1 shows an example of how Azure DevOps will be used to keep track of the project progress.

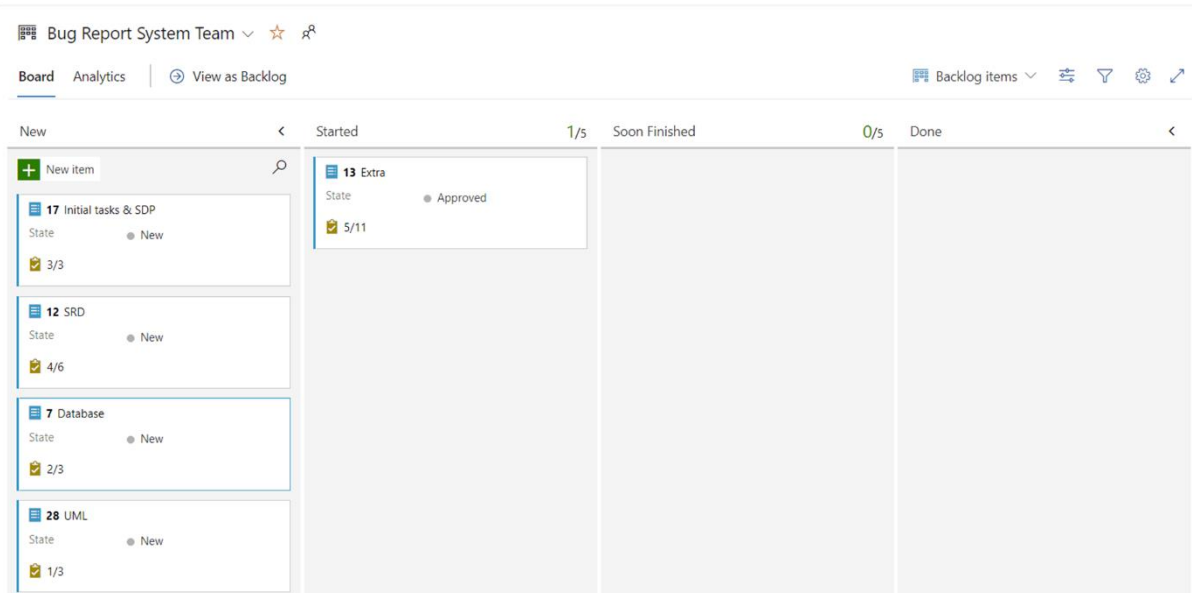


Figure 8-1: Azure Task example

9. Tools

The following list names the (software) tools that will be used for the development of the project:

- MS Visual Studio
- SQL Server Express
- Erwin Database modeller
- MS Teams
- MS Office programs
- MS Visio
- Azure Dev Ops
- [UMLetino - Free Online UML Tool for Fast UML Diagrams](#)