

**Operation Sundew**



November 14, 2019

Console Commander

https://www.consolecommander.net/

# 1. Introduction

We have a problem in which we are constantly using Microsoft Word to show what bugs we have and how we are attempting to solve them. We need something like Jira so that we can log bugs. We don’t’ have the budget for that.

My plan is to create Sundew a bug tracking software that which can contain projects, the people assigned to the project and bugs for each project. So if I work with a team, we can use it to report bugs and solve them together as a group.

This will stop us from having to report bugs verbally and interrupting each other’s work or sending emails back and forth that contain duplicate descriptions about the bug.

As an optional requirement we can allow this system to email other parties upon logging of the bug including the sender as well.

Not just anyone can be able to use this so we will need a login system.

## Purpose

To make life easier for developers to stop from emailing duplicate bug reports to each other. To have a system that show bugs that are resolved and not resolved and have an automated process of emailing developers the bug status until app can be pushed into production or made useful for software development teams.

Purpose is also to make this application reusable by having backend database as reusable as possible for other CRUD business scenarios.

## Intended Audience

Developers, quality assurance members, testers.

## Scope

We are developing bug tracking web software.

### Benefits

* Increased productivity and less interruption.
* Centralized system that teams can tap into especially while remote.
* Allow team to track reported bugs, and comment on each one and be notified via email and application notifications of future progress including the status of the bug.
* Allows team members to prioritize bugs in severity.

Objectives

|  |  |
| --- | --- |
| **Objectives** | **Complete?** |
| To have title about the problem that tells a story in several different information fields, including the date/time/, ID, description, steps to replicate, who’s assigned to it. |  |
| To allow team to assign members to bugs including themselves. |  |
| Can login and out of system. |  |
| Can display information about bug. |  |
| Can write information about bug to database. |  |
| Can have backend information as generic as possible to be reused for other applications. |  |

## 1.4 Definitions and Acronyms

# 2. Overall Description

## 2.1 User Needs

## 2.2 Assumptions and Dependencies

# 3. System Features and Requirements

## 3.1 Functional Requirements

## 3.2 External Interface Requirements

## 3.3 System Features

## 3.4 Nonfunctional Requirements