Personal Issue Tracker Proposal

ATeam-30

April 5, 2020

Contributors:
James Charapata: jcharapata@wisc.edu
Martin Diges: mdiges@wisc.edu
Tyler Johnston: tjjohnston2@wisc.edu, tjohnston@cs.wisc.edu
Mingrui Leng: mleng2@wisc.edu
Alec Lowry: lowry3@wisc.edu

Contents

1 Problem

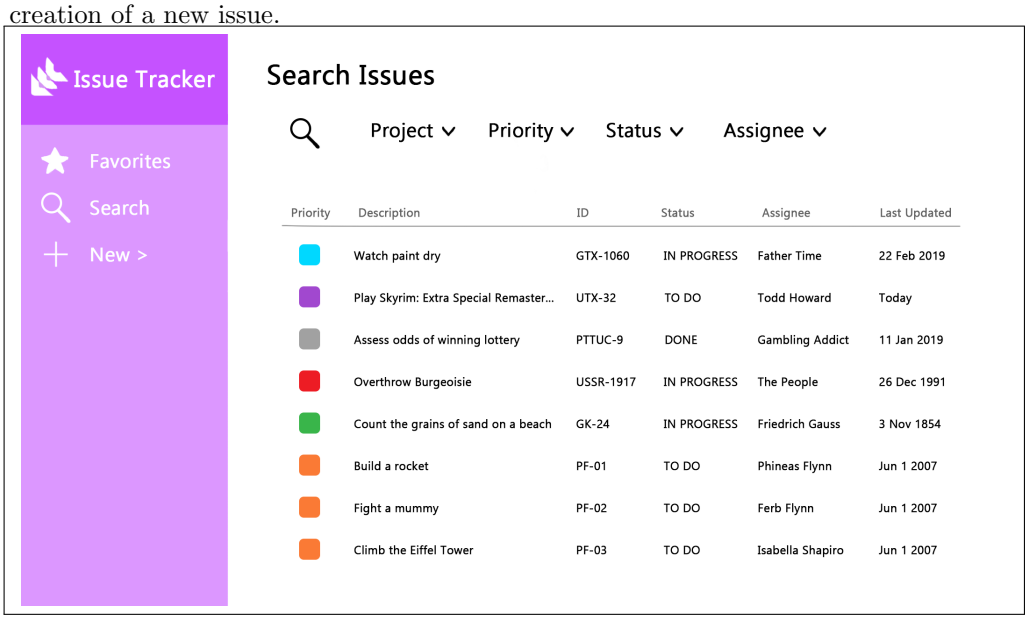
The goal of this product is to offer a comprehensive tool for tracking issues and features to be implemented. Will compliment git version control for easy tracking of the development and implementation process. In addition, it will allow for easy handling of issues that pop up throughout the development process.

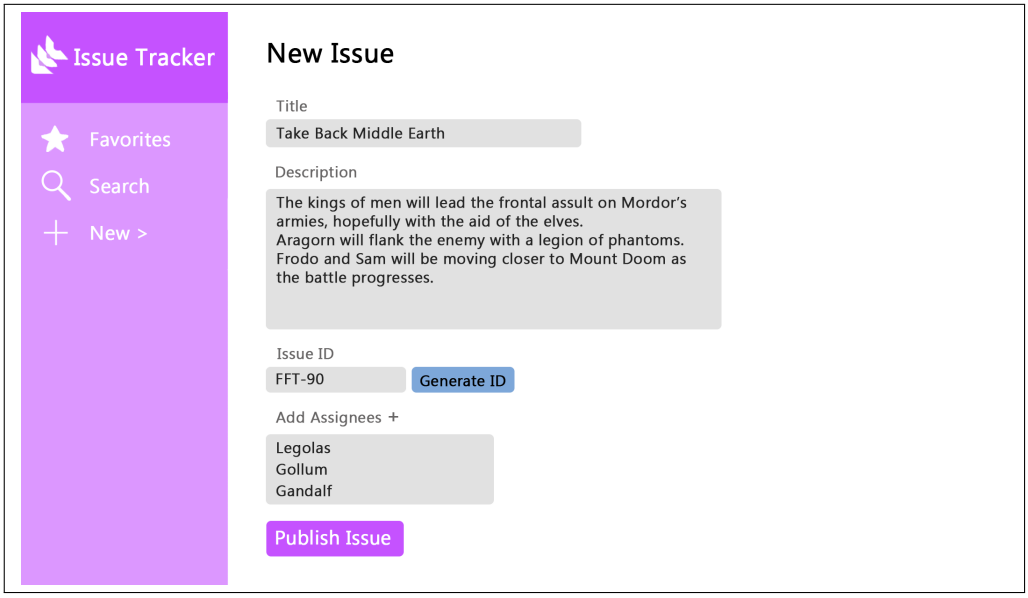
2 Primary Stakeholder

Primarily designed with software development in mind, but could be used for most projects in the development and maintenance phase.

3 Graphical User interface

Upon application launch, IssueTracker initializes to a screen representing all current projects and issues (figure 1). Figure 2 represents a sample screen for the creation of a new issue.





4 Data Structure

Return Type	Method Name	ParameterList	Description
void	Issue()	String name, String description	Creates a new Issue object with a name and description
void	(setters)	Object newValue	Assigns new value to a field
Object	(getters)	-	Retrieves value of a field
void	addAssignee()	String Name	Adds new person who is responsible for completing the Issue
void	removeAssignee()	String Name	Removes person who is responsible for completing the Issue.

Field	Field Type	Description
name	String	name of Issue
description	String	description of Issue
ID	String	ID of Issue
status	Enum (TO DO, IN PROGRESS, DONE)	describes current state of Issue
assignees	ArrayList<String>	people to whom the Issue is assigned
priority	Integer (maybe Double?)	priority of Issue
deadline	Date	date by which Issue will be due
dateCreated	Date	date on which Issue was created
dateLastAccessed	Date	date on which Issue was most recently edited
dateClosed	Date	date on which Issue was closed/completed

Return Type	Method Name	ParameterList	Description
void	newIssue	String Title,String Description, Int Issue ID	Creates a new Issue object with a title, description, and issue ID.
void	editIssue	String Title,String Description, Int Issue ID	Assigns new title, description, and issue ID fields to Issue object
void	removeIssue	String Title	Removes Issue object from issueList
void	sortIssues	String field	Sorts the projects by field, specified by an integer choice parameter.

Fields	Field Type	Description
issueList	ArrayList<Issue>	contains Issue objects
name	String	name of Project
description	String	description of Project
deadline	Date	date by which Project will be due
dateCreated	Date	date on which Project was created
dateLastAccessed	Date	date on which Project was most recently edited
dateClosed	Date	date on which Project was closed/completed
open	boolean	true if open, false if closed/completed

Return Type	Method Name	ParameterList	Description
void	newProject		Create new Project
void	editProject		Edit fields of a project
void	removeProject		Delete a project permanently
void	serializeJSON		Write java objects to JSON
void	deserializeJSON		Create java objects from JSON
array[String]	displayProject		Send all Project data to GUI for display. Might be done automagically by JavaFX
void	sortProjects	String field	Sorts the projects by field, specified by an integer choice parameter.
Field Name	**Field Type**		
projectList - Private	ArrayList<Project>		
(future credentials for acccount owner)? - Private			
(whatever construct is needed to integrate with JavaFX) - Private			

5 Object Diagram