Project Specification Document

John Deere - Standardized Metrics Reporting and Storage

Jordan Meidinger
Jayden Rosenau
Anna Carlson
Wyly Andrews

Table of Contents

Section | Page

1. Project Vision and Objectives 1
1.1 Project Scope and Vision 1
1.2 Project Goals and Objectives 1
2. Project Planning 2
2.1 Project Lifecycle 2
2.2 Project Setup 2
2.3 Stakeholders 3
2.4 Project Resources 3
2.5 Assumptions 3
3. Project Tracking 4
3.1 Tracking 4
3.2 Communication Plan 4
3.3 Deliverables 5
3.4 Project Metrics 5
4. Requirements (User Stories) 6
4.1 Overall Description 6
4.2 Users and Roles 6
4.3 Use Case Diagrams 7
4.4 User Stories (Requirements) 7
4.5 Constraints and Limitations 13
5. Design 14
5.1 Introduction / Scope 14
5.2 High-Level Component Design 14
5.3 Class Diagram 14
5.4 Activity Diagrams 15
5.5 Sequence Diagram 16
5.6 Alternative Designs and Design Rationale 17
5.7 Data Architecture 17
6. User Interface 18
6.1 UI Description 18

6.2 UI Mockup 19

7. Project Closure 20

7.1 Goals / Vision 20

7.2 Delivered Solution 20

7.3 Remaining Work 20

8. Definitions and Acronyms 21

1. Project Vision and Objectives

1.1 Project Scope and Vision

At John Deere, developers are using many different development tracking tools for following the progress of projects and defects. Since there are a lot of information spread out across these different tools, John Deere plans to centralize the information in one place for ease of access. This project should be the development of a REST API and associated database that can serve as a single entrance access to the centralized database in order to store and access the information there. Hopefully by developing this system, important project progress information can be tracked more easily, and assigned tasks towards project team members can be distributed as project teams change size.

The REST API should be developed to handle queries made for the database. A database schema should be created such that defects can be tracked according to engineer, project, review, or other metrics beneficial to the development teams. Documentation about the project and the source code should be maintained throughout the project so that usage, maintenance, and improvements can be done after the project has been completed.

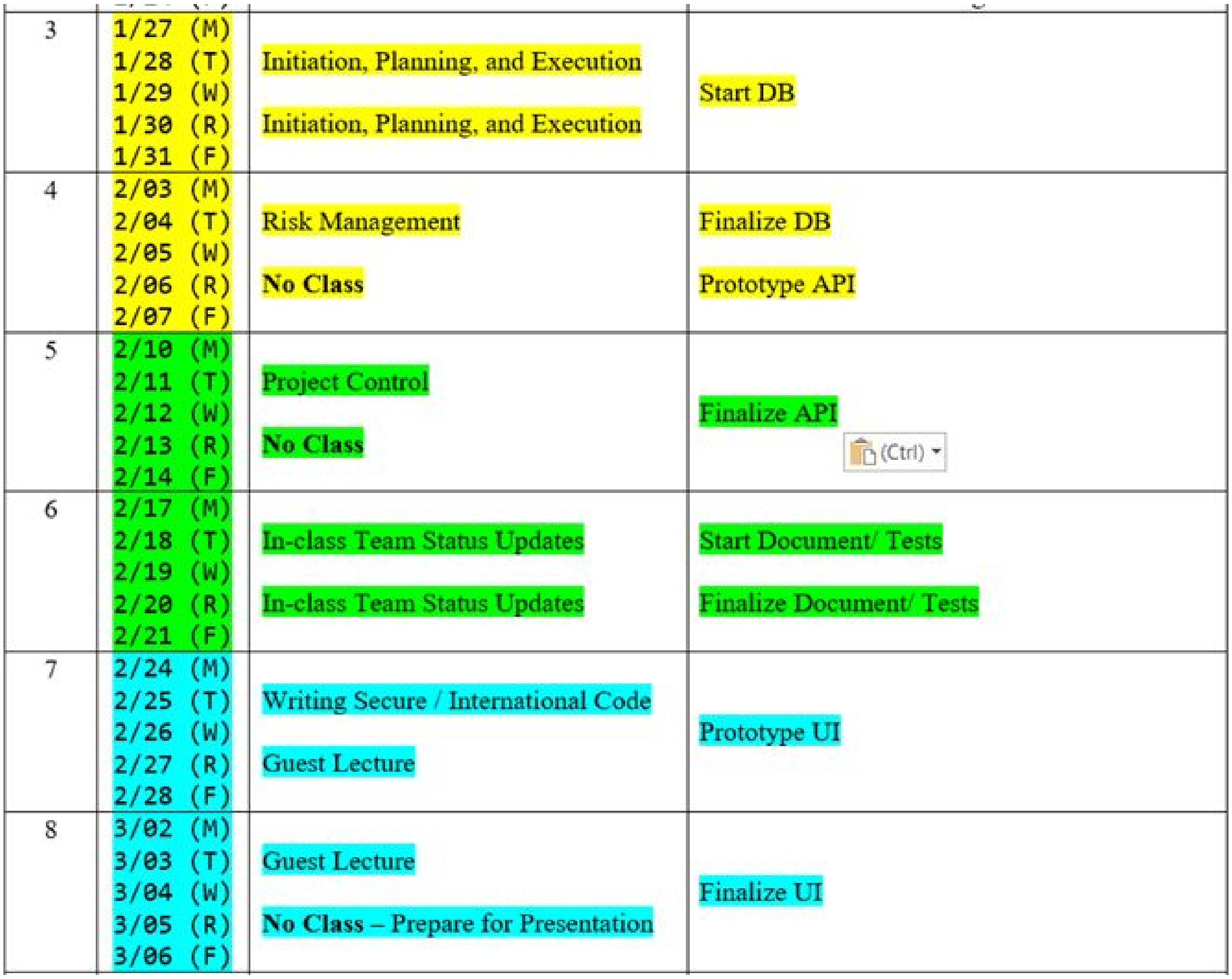
1.2 Project Goals and Objectives

#	Goal or Objective
1	Every group member is efficient in Django.
2	Make the system easy to update and maintain.
3	Website is Secure
4	System is easy to understand and use

2. Project Planning

2.1 Project Lifecycle

Each color represents a sprint, starting with yellow.



3	1/27 (M) 1/28 (T) 1/29 (W) 1/30 (R) 1/31 (F)	Initiation, Planning, and Execution Initiation, Planning, and Execution	Start DB
4	2/03 (M) 2/04 (T) 2/05 (W) 2/06 (R) 2/07 (F)	Risk Management No Class	Finalize DB Prototype API
5	2/10 (M) 2/11 (T) 2/12 (W) 2/13 (R) 2/14 (F)	Project Control No Class	Finalize API
6	2/17 (M) 2/18 (T) 2/19 (W) 2/20 (R) 2/21 (F)	In-class Team Status Updates In-class Team Status Updates	Start Document/ Tests Finalize Document/ Tests
7	2/24 (M) 2/25 (T) 2/26 (W) 2/27 (R) 2/28 (F)	Writing Secure / International Code Guest Lecture	Prototype UI
8	3/02 (M) 3/03 (T) 3/04 (W) 3/05 (R) 3/06 (F)	Guest Lecture No Class – Prepare for Presentation	Finalize UI

2.2 Project Setup

#	Decision Description
1	GitHub for code storage.
2	Django, Plotly.
3	Trello and Google docs for documentation and timeline.

2.3 Stakeholders

Stakeholder	Role
Abram Haich	John Deere Contact (Sponsor)
John Guertin	John Deere Contact (Sponsor)
John Deere Development Team	Potential End Users
Alex Radermacher	Instructor
Dean Knudson	Instructor
Jordan Meidinger	Team Member
Jayden Rosenau	Team Member
Anna Carlson	Team Member
Wyly Andrews	Team Member

2.4 Project Resources

Resource	Resource Description	Quantity
Capstone Team	Our team of students who will be the primary developers of the project.	4
GitHub Account	Used to share code with each other for development	4
Trello Account	Allows team to share files, assign tasks, and stay on schedule.	4
Mock Data	Used to test Django application and other features that are desired	1
Sponsor contacts	The mentors we can contact for assistance and clarification about the project.	2
NDSU CS server	Available for hosting potential mock databases.	1

2.5 Assumptions

#	Assumption
1.	We will meet every week with ether Abram, John, or both.
2.	All software is open source and free to use.
3.	All group members will be able to understand Django and REST api functionality.
4.	Notes/Documentation will be available for each class and function responsibility.
5.	We will be allowed to display in-progress software publicly over GitHub.

3. Project Tracking

3.1 Tracking

Information	Description	Link
Code Storage	Project code and commands will be stored on GitHub	GitHub
Bug Tracking	Bug tracking will be done with GitHub	GitHub
Document and Assignments	Weekly reports, specifications, and design documents will be stored and shared on Trello	Trello
Installation and Troubleshooting	Installation process and FAQ can be found on the read.me on GitHub	GitHub

3.2 Communication Plan

Regularly Scheduled Meetings

Meeting Type	Frequency/Schedule	Who Attends
Onsite meeting	Friday 9-10 Am	Project team, Sponsor
Team meeting	Wednesday Before class	Project team
Sprint planning	Wednesday during team meeting	Project team
Sprint review	Monday during the week after a sprint ends	Project team

Information To Be Shared Within Our Group

Who?	What Information?	When?	How?
Project team	Task and Documentation to be submitted	Weekly	Trello / Google Docs

Information To Be Provided To Other Groups

Who?	What Information?	When?	How?
Sponsor	Progress report	Friday Morning	In person
Sponsor and instructor	Final deliverables	Completion of project milestones	In person, powerpoint, email, Github

Information Needed From Other Groups

Who?	What Information?	When?	How?
Sponsor	Feedback and requirements	Weekly	Email, in person
Sponsor	UI requirements	After second sprint	Email, in person

3.3 Deliverables

#	Deliverable
1	Database Schema
2	REST API
3	Documentation
4	Place to download code
5	
6	
7	
8	

3.4 Project Metrics

Metric	Frequency
Sprint review	Per sprint at end of sprint
Sprint goals	Per sprint at beginning
Effort	Weekly

4. Requirements (User Stories)

4.1 Overall Description

The goal of this project is to make a Rest API for John Deere so that they may keep track of projects from multiple locations and the teams and individuals assigned to them.

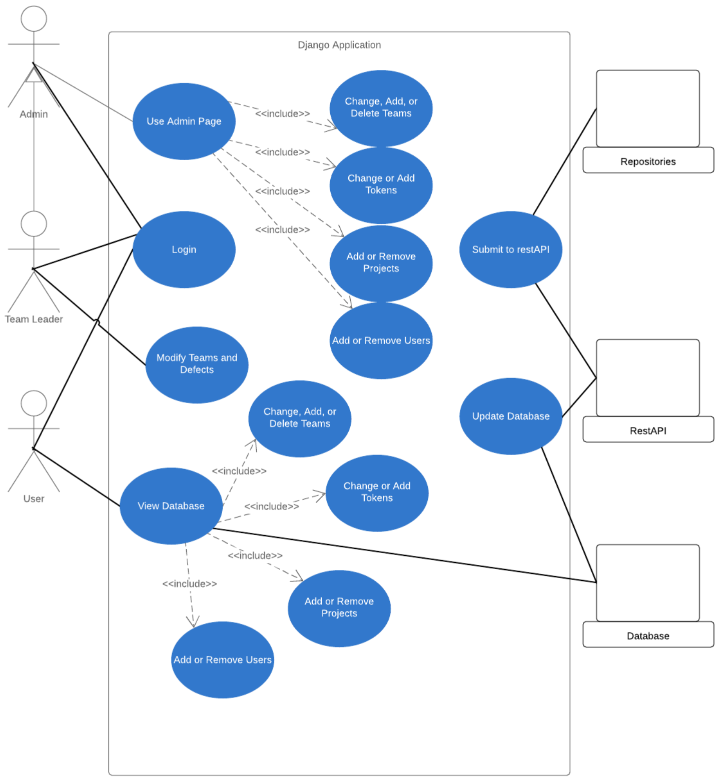
This project requires that we use Django to make the Rest API. We meet on a weekly basis with the sponsors of this project for continued feedback.

Due to the quick time in which our primary task was completed, the scope of this assignment may expand past sprint one to include things such a UI or backwards compatibility with updates to the API affecting the locations they were pulled from. These are considered stretch goals. As such priority is given to necessity components such as the API itself. Additional items are to add to the functionality of that main item or in assisting the users in tasks related to its operation and are not to interfere with it.

4.2 Users and Roles

User	Description
Admin	Allows changes to teams,projects,defects,tags,and tokens.
Standard User	Allowed to view information that is for their specific team or project.
Team leader	Has supervision over their team and projects and can make slight changes to defects and teams.
CEO	Views dashboard for verifications on progress being done.

4.3 Use Case Diagrams



4.4 User Stories (Requirements)

ID	Feature	Story Points
US-01	Setup Rest API	10
US-02	Differentiate Authorization levels	5
US-03	Document how to use system	3
US-04	Setup UI	5
US-05	Incorporate Pivot Graphs based on the info in Rest API	5
US-06	Make changes to Rest API also update original source material	7
US-07	Enable batch data uploads	5
US-08	Setup dashboard for user to quickly view relevant info to them	5
US-09	Enable user to alter personal dashboard and save settings	5
US-10	Add project specific graphs to the project details pages	5
US-11	Documentation	3

SPRINT 1

Total Estimated User Story Points for Sprint 1: 30 points
Actual Completed User Story Points for Sprint 1: 30 points

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
US-01	2/08/20	As an admin I want to have access to an database so I can keep track of the moving parts of a complex project and the players involved with it.	**commit**	**10**	**10**	**100**

Acceptance Criteria		Verification
1.2	Get information from api	Showed John Deere example script.
1.3	Post information into the api	Showed John Deere example script.
1.4	Delete information from the api	Showed John Deere example script.
1.5	View information on dashboard	Showed John Deere example script.
1.6	Use auth tokens.	Showed John Deere example script.
1.7	Remove auth tokens	Showed John Deere example script.

ID	Tasks	Resource
1.A	Setup Python Env	Jordan
1.B	Setup GitHub	Jordan
1.C	Setup Django Base	Jordan
1.D	Create Database Schema	Sponsor
1.E	Create new django app (Rest API)	Jordan
1.F	Create new django app (Account)	Jordan
1.G	Develop UI	Anna
1.H	Test requests	Jayden
1.I	Documentation	Team

SPRINT 2

Total Estimated User Story Points for Sprint 2: 20
Actual Completed User Story Points for Sprint 2: 20

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
US-03	2/18/20	As an admin, I would like the project and all related components to be well documented so that the project can be expanded upon once the capstone team is done.	**commit**	**3**	**3**	**100%**

	Acceptance Criteria	**Verification**
3.1	The database models and views are well documented and have enough information to create new tables	The project sponsor approves the documentation that we present
3.2	All process and functions are explained clearly so someone new can understand how to run them	The project sponsor approves the documentation that we present

ID	Tasks	Resource
3.A	Document models	Team
3.B	Document views	Jordan / Wyly
3.C	Document test scripts	Team
3.D	Document process	Team

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
US-04	2/18/2020	As a standard user I want to use a UI to view the info in the rest API so that I don't need to understand python to use it.	**Open**	**5**	**5**	**100%**

	Acceptance Criteria	**Verification**
4.1	A mock-up site will be made using	JDES sponsor examines it and approves

	HTML/CSS to illustrate basic layout plans	
4.2	UI site is made in Django	The original mock-up can be disposed of and the Django version can be used in minimally the same way
4.3	UI site changes based on test data	When the site displays an ability to change based on a set of date.
4.4	UI site changes based on data in rest API	JDES sponsor approves of UI in site.

ID	Tasks	Resource
4.A	Make mock-up site	Anna
4.B	Apply changes to UI based on test data	Team
4.C	Connect UI up to rest API	Jayden

SPRINT 3

Total Estimated User Story Points for Sprint 3: 20
Actual Completed User Story Points for Sprint 3: 20

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
US-08	3/27/20	As a user I want to personalize my dashboard for my projects	**commit**	**5**	**5**	**100**

	Acceptance Criteria	Verification
	Easily UI allows users to understand how to filter and choose what favorites are chosen	The project sponsor approves the implementation that we presented

ID	Tasks	Resource
8.1	Allowed filters for open projects or closed	Jordan
8.2	added graphs	Wyly
8.3	filter graphs for dates	Jordan / Wyly

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
US-09	4/20/20	As a user I want to save my favorite projects for my dashboard	**commit**	**5**	**5**	**100**

	Acceptance Criteria	Verification
	Favorite section that allows you to choose what projects that you want to highlight	The project sponsor approves the implementation that we presented

ID	Tasks	Resource
9.1	Add Favorite Section	Jordan / Jayden
9.2	Allow to delete favorites	Jordan / Jayden
9.3	Allow to add favorites	Jordan / Jayden
9.4	Add favorite from other projects	Jordan / Jayden

SPRINT 4

Total Estimated User Story Points for Sprint 4: 8
Actual Completed User Story Points for Sprint 4: 8

<table>
<tr><th>ID</th><th>Added</th><th>Description</th><th>Status</th><th>Story Points</th><th>Actual Equivalent Story Points</th><th>% Completed</th></tr>
<tr><td>US-10</td><td>4/20/2020</td><td>As a user I want to view project specific information from graphs when viewing the project details page</td><td><b>commit</b></td><td><b>5</b></td><td><b>5</b></td><td><b>100</b></td></tr>
<tr><th colspan="3">Acceptance Criteria</th><th colspan="4">Verification</th></tr>
<tr><td>10.1</td><td colspan="2">Create project-specific graphs appear on the projects detail page</td><td colspan="4">Containment pie chart and defects per phase graph both appear on the projects details page with valid information</td></tr>
<tr><th>ID</th><th colspan="4">Tasks</th><th colspan="2">Resource</th></tr>
<tr><td>10.A</td><td colspan="4">Update the graph functions to make database pulls for specific projects</td><td colspan="2">Wyly</td></tr>
<tr><td>10.B</td><td colspan="4">Add to the template a location for the new graphs</td><td colspan="2">Wyly</td></tr>
</table>

<table>
<tr><th>ID</th><th>Added</th><th>Description</th><th>Status</th><th>Story Points</th><th>Actual Equivalent Story Points</th><th>% Completed</th></tr>
<tr><td>US-11</td><td>4/29/2020</td><td>As a user I want to be able to read documentation on the project</td><td><b>commit</b></td><td><b>3</b></td><td><b>3</b></td><td><b>100</b></td></tr>
<tr><th colspan="3">Acceptance Criteria</th><th colspan="4">Verification</th></tr>
<tr><td></td><td colspan="2">Be able to start and change the project</td><td colspan="4">The sponsor reads the rough draft</td></tr>
<tr><th>ID</th><th colspan="4">Tasks</th><th colspan="2">Resource</th></tr>
<tr><td>11.1</td><td colspan="4">How to start the project</td><td colspan="2">Jordan</td></tr>
<tr><td>11.2</td><td colspan="4">How to update the project</td><td colspan="2">Jordan</td></tr>
<tr><td>11.3</td><td colspan="4">How to remove items from the server</td><td colspan="2">Jordan</td></tr>
</table>

4.5 Constraints and Limitations

Constraint	ID
One sponsor was gone for a week	CL-1
Covid - 19 Setback	CL-2

5. Design

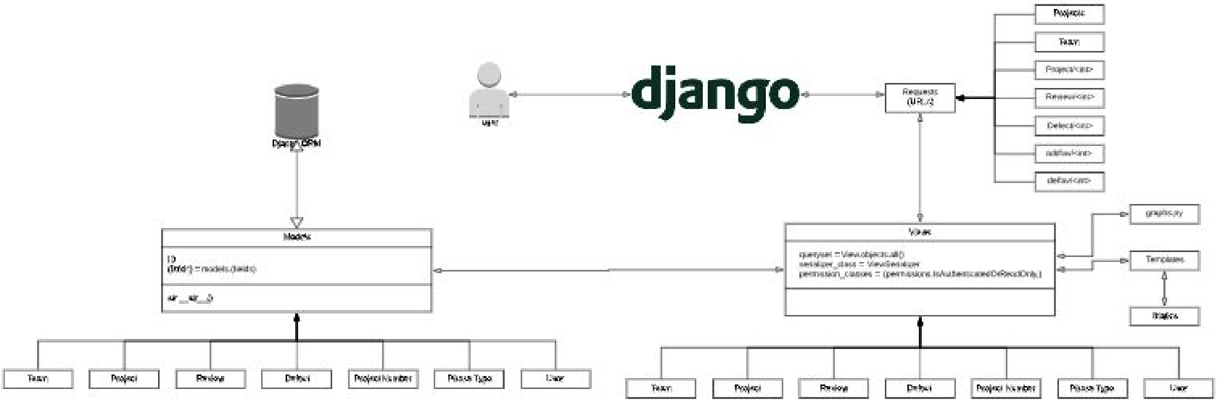
5.1 Introduction / Scope

This project is a free program for the John Deere Corporation. It will be used by the Engineers of multiple projects, teams ,and divisions among John Deere. SMRS will achieve data collection, storing, and filtering defects and reviews from all projects. It stores this information after retrieving it from the used repositories such as Github and Collaborator. After successfully collecting and storing the data the program allows for filtration of the reviews and defects pertaining to the user logged in. This application will be used locally for the time being but with potential of being implemented in different cities.

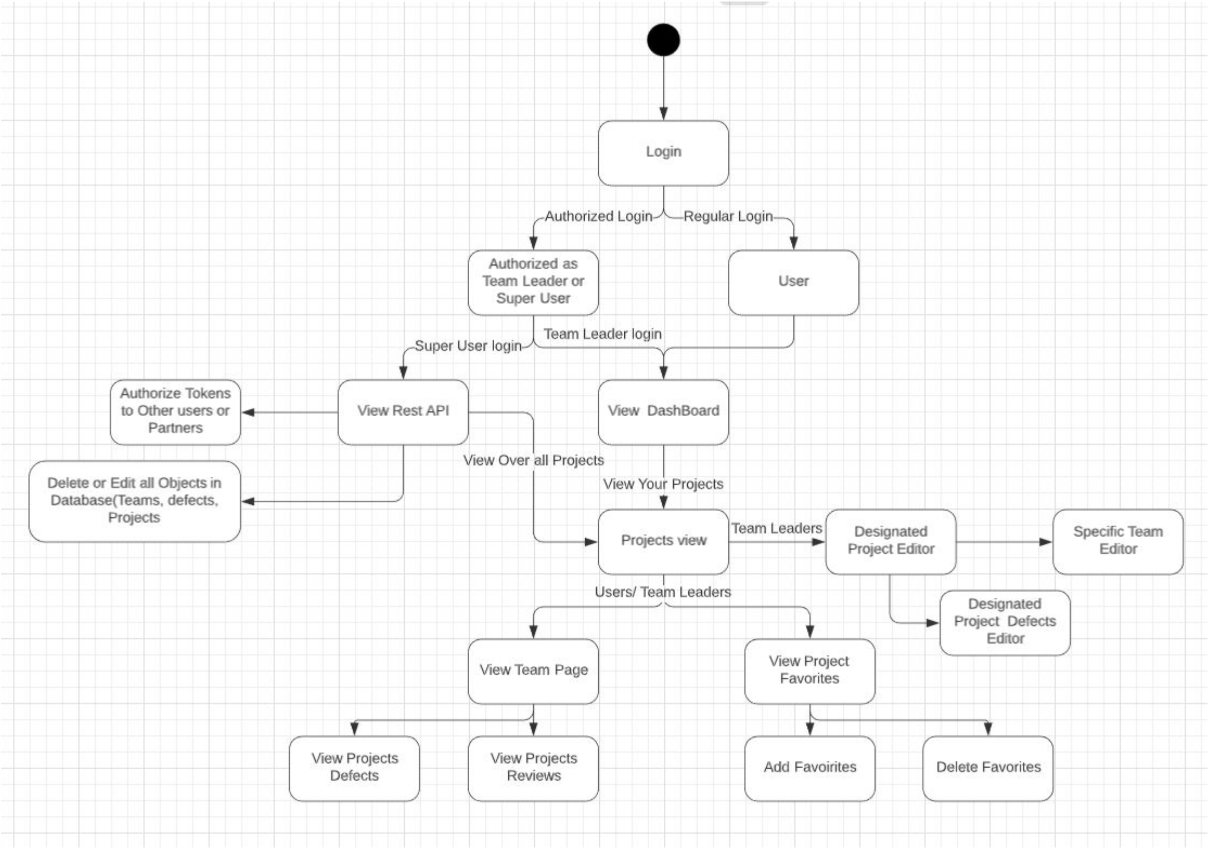
5.2 High-Level Component Design

Component	Related Requirements	Description
Account	Team/Groups	Used to authenticate user and store user information.
RestAPI	Rest Framework	Takes DB information to create urls and views for the django models.
Dashboard	Plotly	Plots x and y values to bar graphs, line graphs, or tables for users to understand large amounts of data.

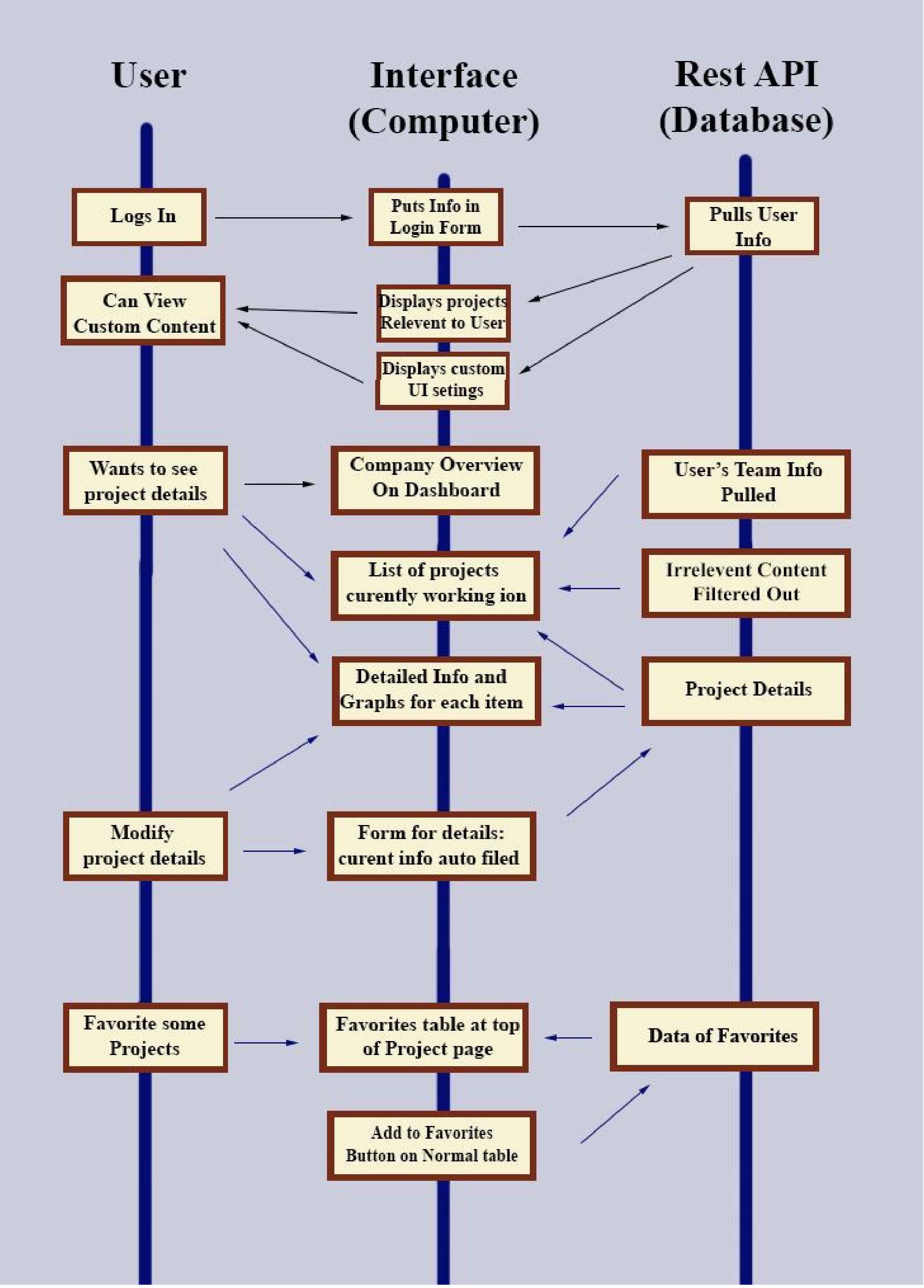
5.3 Class Diagram



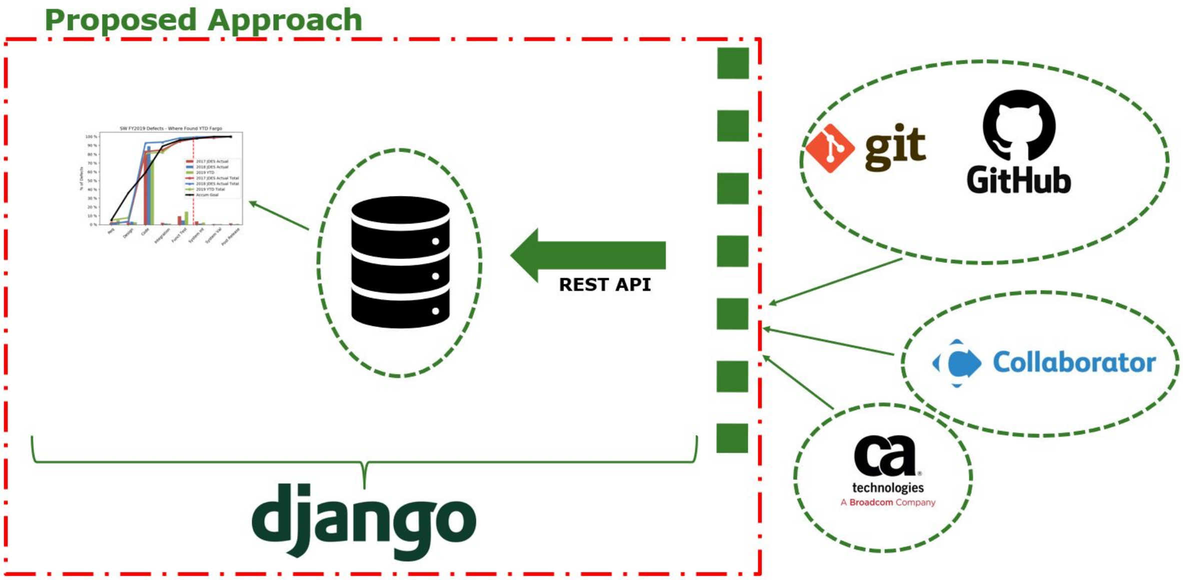
5.4 Activity Diagrams



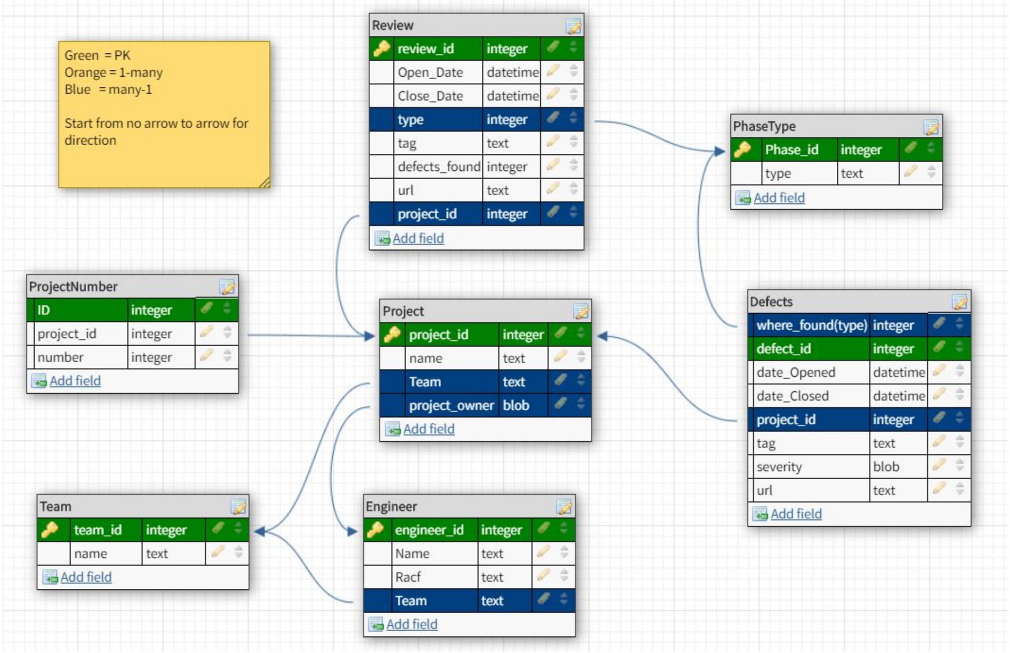
5.5 Sequence Diagram


User
Interface (Computer)
Rest API (Database)
Logs In
Puts Info in Login Form
Pulls User Info
Can View Custom Content
Displays projects Relevent to User
Displays custom UI setings
Wants to see project details
Company Overview On Dashboard
User's Team Info Pulled
List of projects curently working ion
Irrelevent Content Filtered Out
Detailed Info and Graphs for each item
Project Details
Modify project details
Form for details: curent info auto filed
Favorite some Projects
Favorites table at top of Project page
Data of Favorites
Add to Favorites Button on Normal table

5.6 Alternative Designs and Design Rationale



5.7 Data Architecture



Green = PK
Orange = 1-many
Blue = many-1

Start from no arrow to arrow for direction

Review

review_id	integer
Open_Date	datetime
Close_Date	datetime
type	integer
tag	text
defects_found	integer
url	text
project_id	integer

PhaseType

Phase_id	integer
type	text

ProjectNumber

ID	integer
project_id	integer
number	integer

Project

project_id	integer
name	text
Team	text
project_owner	blob

Defects

where_found(type)	integer
defect_id	integer
date_Opened	datetime
date_Closed	datetime
project_id	integer
tag	text
severity	blob
url	text

Team

team_id	integer
name	text

Engineer

engineer_id	integer
Name	text
Racf	text
Team	text

6. User Interface

6.1 UI Description

Dashboard

The UI is set up in such a way that a busy person should be able to get a great deal of information at a glance. The main page shows some common large scale metrics that you can scroll down in to get a bigger picture.

The main items are arranged so that you can view each item by scrolling down. While the current plan is to keep this on a desktop, this would allow for an easier shift to a smartphone if that route is desired in the future. Information is by default in the range of the last three years, setting the start and end dates by the business year. The user is given the date range at the top, so they don't have to hunt for it.

Projects

There is a section for information on individual projects. If the user is not logged in he or she can use the filter drop down to search for the desired item, and if the user is logged it the projects will automatically be pulled from his or her team.

When the project is open the user can further see more information on reviews and defects. He or she can also scroll down slightly to see some glance-over chars: Number of Defects Per Release, and Post Release Containment.

If the user further clicks on a review or defect he or she is taken to a page where he or she can view and update the individual items information.

Login only Options

Some items are only relevant to users, and will only pop appear as an option after the user has logged in.

User

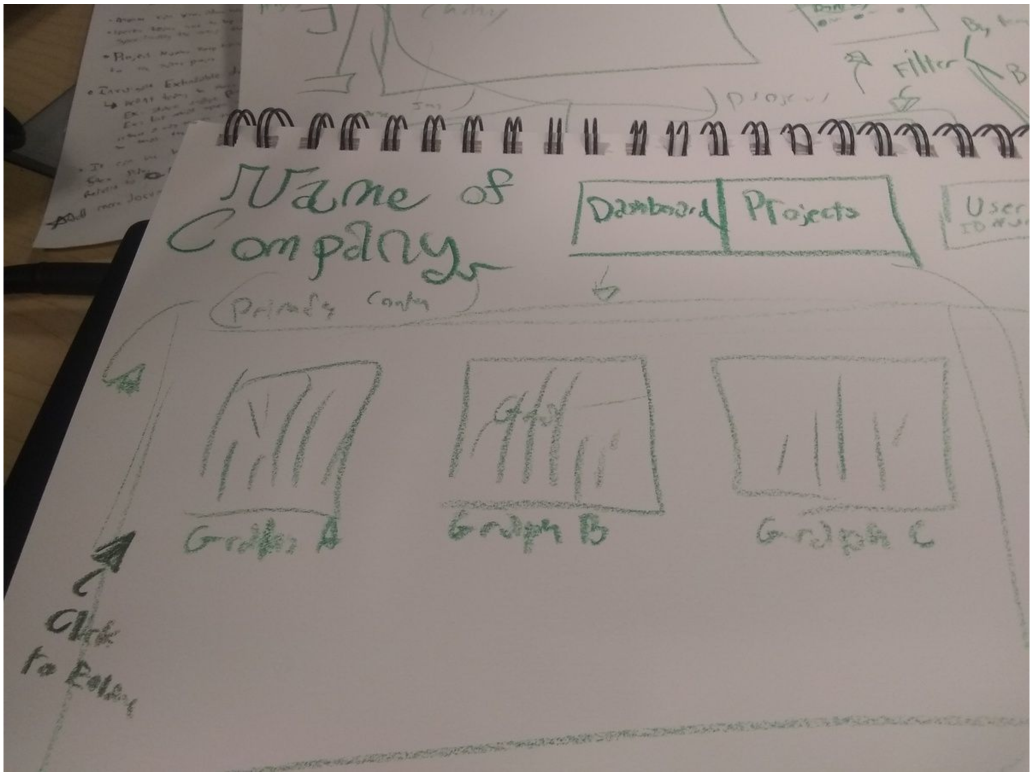
Here the user can change his or her displayed name, change his or her team, or change the theme.

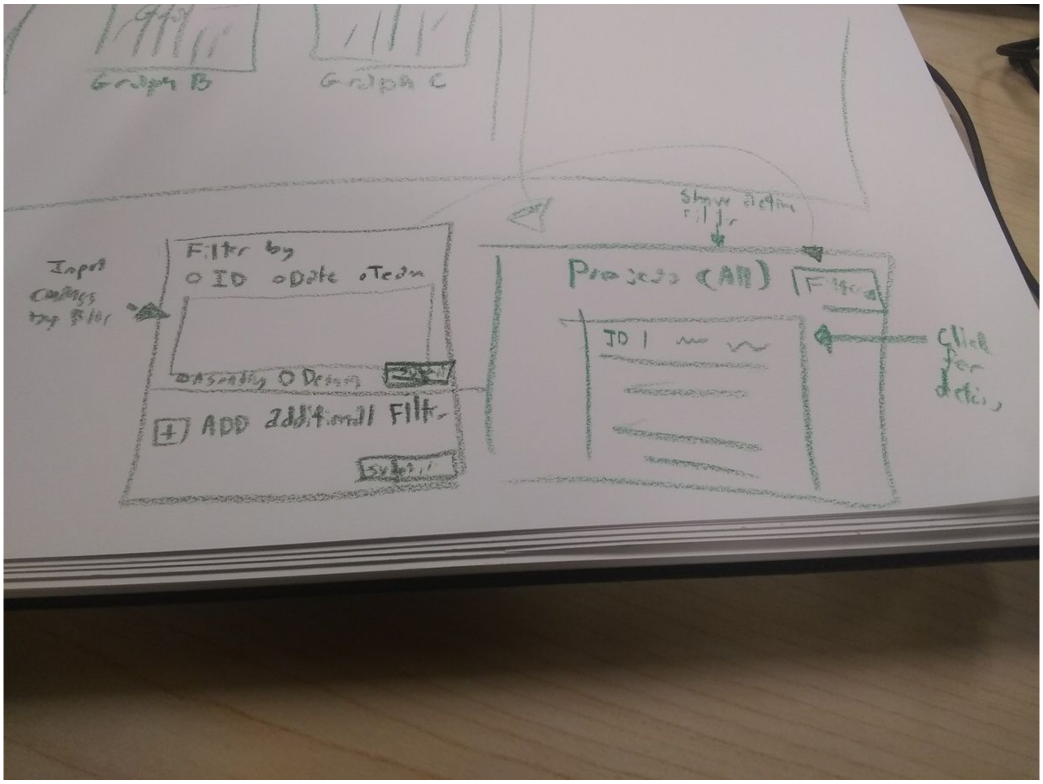
There are options for light and dark mode, so that the user may switch between a clean, professional look in white, and an easy on the eye dark mode for those who stare at their screens longer.

Team

The team section will display the other members of the team as well as the current projects. Clicking on one of these will send the user to that project's information.

6.2 UI Mockup


Name of Company
Dashboard
Projects
User ID #
Primary Content
Graph A
Graph B
Graph C
Click to Return


Graph B
Graph C
Show active filter
Input changes by filter
Filter by
○ ID ○ Date ○ Team
○ Ascending ○ Descending
+ ADD additional Filter
Submit
Projects (All)
Filter
ID 1
Click for details

7. Project Closure

7.1 Goals / Vision

This project will collect and store the data from outside repositories.The program allows for filtration of the reviews and defects pertaining to the user logged in. This application will be used locally for the time being but with potential of being implemented in different cities.

7.2 Delivered Solution

https://github.com/jwmeidinger/SMRS

Also sent the project in Zip with all of the other documentation.

7.3 Remaining Work

1. Set up a server.
2. Add users.
3. Create a script to get info from Github and other development tracking websites and use the REST API to input that info into the database.

8. Definitions and Acronyms

Term	Definition
SMRS	Standardized Metrics Reporting and Storage
API	Application programing interface - A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.