



Bilkent University

Department of Computer Engineering

CS 353 Term Project

Project Tracking Software

Proposal Report

Bartu Atabek 21602229

Utku Görkem Ertürk 21502497

Hygerta Imeri 21603212

Deniz Yüksel 21600880

Instructor: Özgür Ulusoy

Assigned TA: Aytek Aman



March 04, 2019

This report is submitted (softcopy) to <http://track-it.cf> in partial fulfilment of the requirements of the Term Project of CS353 course.

Table of Contents

Table of Contents	1
1. Introduction	3
2. Project Description	3
2.1 Audience Specifications of the system:	4
3. Why a database is going to be used as a part of the system	5
3.1 How a database is going to be used as a part of the system	5
3.1.1 User creation	5
3.1.2 Login	6
3.1.3 Trivial database actions	6
3.1.4 Statistics	6
3.1.5 Scheduler	6
3.1.6 Report generator	6
4. Requirements	6
4.1 Functional requirements	6
4.1.1 Basic Collaborator Capabilities	7
4.1.2 Subscribed Collaborator Capabilities	7
4.1.3 Basic Admin Capabilities	7
4.1.4 Subscribed Admin Capabilities	8
4.2 Nonfunctional Requirements	8
4.2.1 User-friendliness	8
4.2.2 Response time	8
4.2.3 Maintainability and Reliability	8
4.2.4 Notification System	9
4.3 Pseudo Requirements (Constraints of the System)	9
5. Limitations	9
5.1 System Limitations	9
5.2 Technical Limitations	10
5.2.1 Variety of Data	10
5.2.2 Data Transaction Speed	10
5.2.3 Data Size	10
5.2.4 Security	10
5.2.5 Reliability and Stability	10
5.2.6 Limitations of Relational Languages	10
5.2.6.1 Server Limitations	11

5.2.7 Maintainability	11
6. Recommended Hardware and Software Specifications	11
7. Entity-Relationship (E/R) Diagram	12
8. Website Link	13
9. Conclusion	13
10. References	13

1. Introduction

This report is a proposal to a client web-application of an issue-tracking software which incorporates a database system. This application will have several types of users that will be able to schedule, assign and complete tasks in projects that are opened by other users. As the name of this application suggests, "TrackIT" is used in tracking issues of projects using information technologies (IT). In the following subsections of this report, Its main functionalities and mechanics will be described. Furthermore, the requirements needed for the project to fulfill its tasks will be analyzed and the technical and system limitations will be argued. Moreover, we will state why our web application will need a database system, and how the database system will be utilized to implement TrackIT. We will also include the E/R diagram that we came up with for this project. Finally, a link to our webpage including an online version of this report as well as information about team members will be provided.

2. Project Description

TrackIT is a general issue tracking system. This tool is a client web application which can be used by teams in collaborative settings and by individuals to track the development of personal projects (deadlines, tasks, etc.). A newly-designed browser-based user experience provides users to connect from different platforms through a cross-browser compatible interface therefore avoiding the installation of any extra application. Registration to the system is free and the newly registered user will be able to access all the standard features of the project. On the other side, subscriptions are supported and they allow users to use some other premium features which will be explained in detail in the following sections of this report. Besides the basic functionalities of raising issues to other team members and creating personal and team-wise tasks, TrackIT is thought to provide users with some analytical reports and statistical results which will be generated by the system

upon user request. Making use of the system requires only one account which can be created by the user in a few simple steps. Each user will be able to create an account and fill in their personal information which will be stored in the systems' database as firstly provided by the user. Similarly, the user will need to make some theme-related decisions which can be customized later on through the settings webpage. The settings page allows users to change their personal data.

2.1 Audience Specifications of the system:

TrackIT users can be standard or privileged. These two user types of the issue-tracking system will differ in the features they will modify the project. Each user who can be an admin and/or collaborator to one or more projects has a specific role in a team/project (admin/collaborator) which affects his/her access towards some of the features of the project. For example, admin is the creator of the project and the only one who is allowed to delete it. Still, all other collaborators can track issues assigned to other team members, can assign issues themselves and other team members.

TrackIT supplies the user with a schedule-uploading option which allows the user to note down agenda activities. Premium users can have the system generate some possible meeting time slots between team members. Users will be able to attach files to tasks and projects. In the meantime, the user can generate a 'release-version' attachment from the database which will be stored in the database if needed for future use.

TrackIT can be easily adopted by users to keep track of personal projects but it mostly favors teams in the way it manages the workflow of the project between members, in the way it points out the approaching of a deadline, let it be the end date of a task or a project. The kind reminders tend to avert any delay in the completion of the tasks and projects. The notification system grants the fact that each team member gets notified whenever they are assigned an issue from another team member. All issues of a project can be seen by all team members which are allowed to reassign an issue to themselves. This would indicate a change/update in the team members' issues board (in his/her homepage).

TrackIT saves all the data but also keeps sensitive data confidentially stored and not shared with third parties.

This project is always opened to and appreciates feedback and further improvement suggestions from the users. For this reason, the interface provides the user with a suggestion box to share their thoughts about their experience using TrackIT.

3. Why a database is going to be used as a part of the system

TrackIT is built as a web application that will be used to organize teamwork collaboration. The application includes projects, teams, issues, and release tracking. It also has an account system that keeps each user's data separate. Therefore our web application will need a database because we need to keep a significant amount of data. Because this application is not static but dynamic, application data cannot be embedded into the code itself.

There are two main different structures to store data: File-based data storage and database based data storage. File-based storage is faster for I/O operations but better for small systems, in other words, systems for small data storage requirements. On the other hand, the database system is slower for I/O operations but better for large scale systems, in other words, systems for high data storage requirements. TrackIT is designed for large scale web application so that the database system is chosen as the structuring of data.

3.1 How a database is going to be used as a part of the system

As seen in the E/R diagram representation, this database is used to separate and keep different entity sets and their relationship sets. This separation of data eases modularity, security, and maintenance cost. The paragraph above described some core usage of database further explained as follows:

3.1.1 User creation

When a user fills the signup form and submits it, data of the submit form kept in the user table of the database.

3.1.2 Login

When a user login to the system server side of the application retrieves data from the database and validate the login credentials.

3.1.3 Trivial database actions

Creation, modification, deletion of the project, issue, releases, theme/style, tag, schedule, task, attachment, team - This part is trivial (actions are self-explanatory) therefore its explanation is omitted.

3.1.4 Statistics

By retrieving start and end time of issues, releases and projects, overall statistics will be derived.

3.1.5 Scheduler

All schedules from team retrieved from the database(from Schedule entity creation table) and available times for all desired members shown.

3.1.6 Report generator

This section requires multiple information retrieval as same as statistic subsection. The server side of the application retrieves releases and project. By parsing data from them, it creates reports and by using the attachment table it keeps in the project.

Note: Detailed functionality will be explained in the following section.

4. Requirements

4.1 Functional requirements

There are two types of subscriptions. The actions of a standard user are also supported for premium users. However, premium users have more advantages and

use cases as well. Meanwhile, a user (either premium or standard) can be the admin of a project or a collaborator. Functionalities that admins and collaborators can utilize later.

4.1.1 Basic Collaborator Capabilities

A basic collaborator can utilize the following functionalities:

- Assign a task to a member of the same team,
- Attach a tag to a task,
- View a task in detail,
- Open an issue to another member in the same team,
- Attach a tag to an issue,
- Select from a variety of color themes for the application,
- Create and arrange a schedule for oneself,
- Post an attachment to a task,
- Change the priority of a task,
- Add a note to a task,
- Change the status of a task,
- Check notifications sent by the system,
- Request a release that contains a document and the project code.

4.1.2 Subscribed Collaborator Capabilities

In addition to basic subscribed collaborators, a premium collaborator can use the following functionalities:

- Use scheduler to find common free time between other collaborators and themselves,
- View overall statistics,
- Export a report for overall or desired status,
- Choose from exclusive color themes

4.1.3 Basic Admin Capabilities

In order for a basic user to be the admin of a project, one needs to create a project. After the project is created one automatically becomes the admin and gains the following extra functionalities of a collaborator:

- Dismiss a collaborator,
- Add collaborators,
- Assign roles to collaborators,
- Can re-open an issue and redirect it to another collaborator.

4.1.4 Subscribed Admin Capabilities

An admin that has a premium subscription has the same functions of a subscribed collaborator. Moreover, a premium admin can also utilize the functions that a basic admin can make use of.

4.2 Nonfunctional Requirements

4.2.1 User-friendliness

The web application should be user friendly and easy to use. The UI buttons will be large and visible enough for the easy task and tag creating. Drag and drop feature will be used while creating and assigning tasks or opening issues. The design will be simple as possible and actions will require simple movements to perform. The users will be able to view the projects and tasks in a grid view with a coordinated fashion.

4.2.2 Response time

Although this application is not a competitive game that needs to have immediate responses, TrackIT will provide a response time that is less than 2 seconds while logging in and in-app actions such as opening a task, creating a task and opening an issue.

4.2.3 Maintainability and Reliability

Our database system should be reliable in terms of lack of data leakage. TrackIT should also be scalable for future developments.

4.2.4 Notification System

After a user is assigned a task, a notification will be sent to the user to report the incidents. The user will be able to see the notification whence one launches the web application.

4.3 Pseudo Requirements (Constraints of the System)

- PostgreSQL is going to be used for database implementation of the system,
- PHP is going to be used for the back-end development of the system,
- HTML, CSS, JQuery and Javascript, Bootstrap, Design Paradigms (Material Design etc..) and AJAX is going to be used for the front-end of the system.

5. Limitations

5.1 System Limitations

- Standard users cannot schedule meetings.
- Standard users cannot access to team statistics.
- Standard users cannot export project reports.
- Project meeting scheduling can only be done by the premium users and a minimum of two premium users required to schedule a meeting.
- A project cannot be released if it contains any remaining tasks or issues.
- Task editing and completing can only be done by the project owners, administrators and the user assigned with the task.
- Issue editing and completing can only be done by the project owners, administrators and the user assigned with the task.
- Duplicate Attachments/Tasks/Issues with the same properties cannot be added.
- Assigned tags must be unique within each project.

5.2 Technical Limitations

5.2.1 Variety of Data

The type of database we use for our design constraints us to be careful about the structure of the data we use. The data stored must be structured.

5.2.2 Data Transaction Speed

The speed of the data we process is also constrained to the functionality of the database. In order to overcome this limitation, we will need to provide extra functionality using efficient Information Retrieval techniques.

5.2.3 Data Size

This technical limitation also refers to the scalability potential of the system. Our ability to support a large amount of data is limited to the constraints of the PostgreSQL database.

5.2.4 Security

Each user must log in to the system with a valid account, therefore, each action of the different user type must be well defined and the limitations at the user level related to the user role must be done after the login. The sensitive data stored in the database is confidentially stored and not shared with third parties and is not available through the client side.

5.2.5 Reliability and Stability

If errors occurred during the run time of the system, the elapsed time of fixing operation must be minimized and handled at most network speeds.

5.2.6 Limitations of Relational Languages

Maximum theoretical size of the PostgreSQL is 64tb however, the setup used for this web application is much less capacitated it could achieve a maximum of 10GB

of size. If we considered the system would have many users and therefore exceedingly large data, the probability of an error/mismatch issue increases with relational databases. It's a limitation to be considered if we assume that the TrackIT system reaches a high user count, i.e. used by many.

5.2.6.1 Server Limitations

Maximum sizes PostgreSQL Server (64-bit): 900 bytes per foreign key and 900 bytes per primary key.

5.2.7 Maintainability

The system should be maintainable long-term without causing any major issues and should be easy for other programmers to understand and contribute to it. The documents such as project proposal and design reports will be available to help maintenance personnel to revise or enhance the implementation.

6. Recommended Hardware and Software Specifications

Because it is a web application, minimum requirements are the same as executing a browser with an internet connection. However, recommended and supported environment can be given as:

Hardware:

- Mobile, tablet or desktop environment
- Keyboard, mouse or any pointing device
- 1 MBps or higher internet connection

Browsers:

- Current Safari version
- Current Chrome version
- Current Edge version
- Current Mozilla Firefox version

7. Entity-Relationship (E/R) Diagram

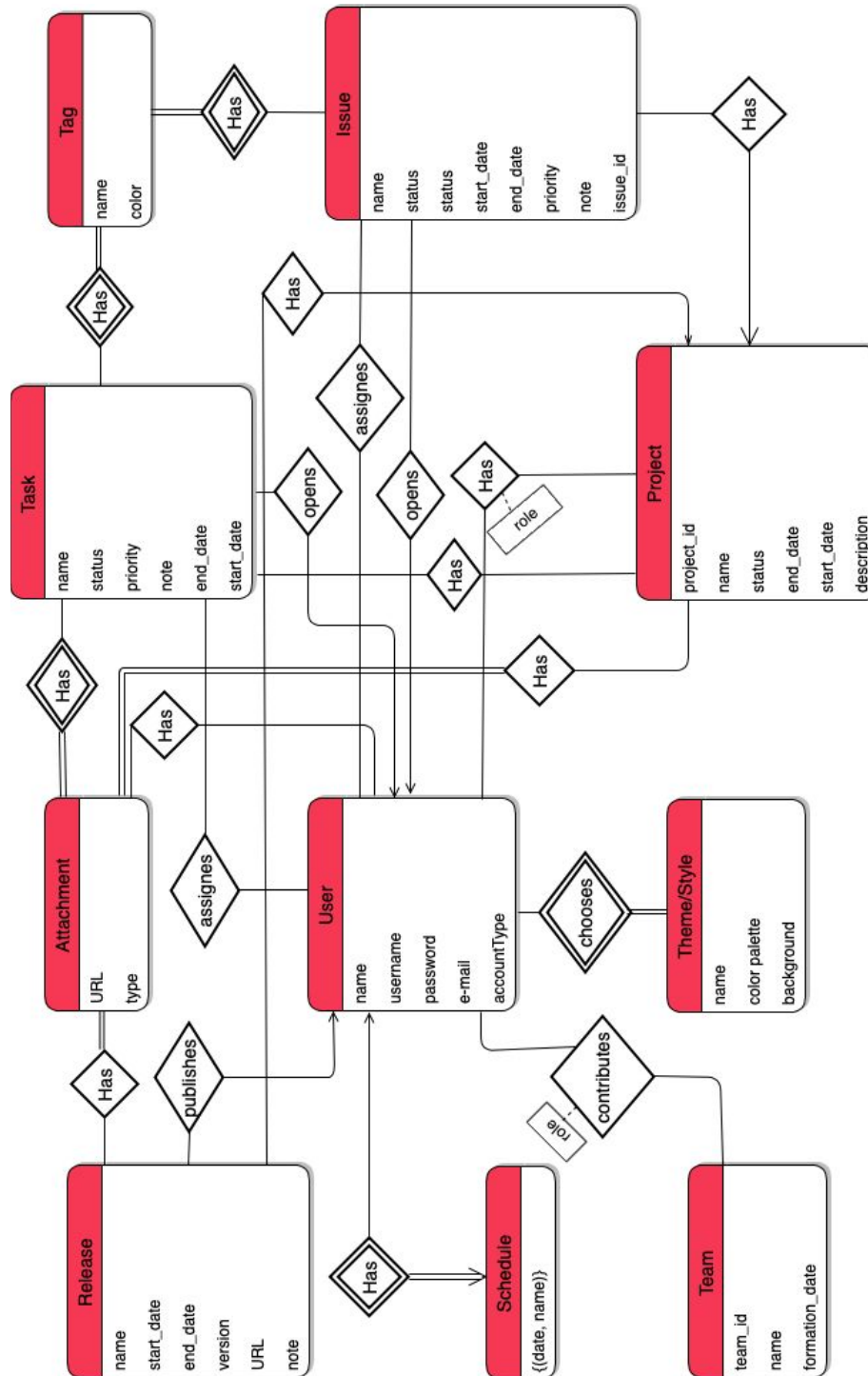


Fig.1. Entity Relationship Diagram of the designed system (TrackIT)

8. Website Link

This report is submitted (softcopy) to <http://track-it.cf> in partial fulfillment of the requirements of the Term Project of CS353 course.

9. Conclusion

In this report, we proposed our database web application TrackIT. While coming up with an E/R diagram we also specified requirements and set constraints by stating limitations. We tried to describe our system briefly but effectively. A link to our website that we will publish further reports and release the project can be found below.

10. References

- [1] <http://atlassian.com/jira>
- [2] <https://www.meistertask.com/app>
- [3] <https://trello.com>