**REQUIREMENTS ANALYSIS DOCUMENT**

1. **Introduction**
   1. **Purpose of the System**

Most of the people work on multiple tasks every day in their job, school or even in home. Working on these tasks is getting harder if you try to handle it with a pen and paper. Also for the people who are responsible for managing a team, it is really hard to monitor the project, tasks and the team. The purpose of the task management system Tasky is making easier and more efficient working and managing the projects and tasks. Users will easily create, assign and prioritize tasks, set deadlines, track how much time spent on tasks and visualize the projects and tasks with the Tasky.

* 1. **Scope of the System**

Task management system Tasky is designed for making tracking a project easier and more efficient. The system consists of a web site and mobile application. Users can create projects in the web site and add participants to the project. Once the project is created users can create tasks within the project. Each task has a reporter and assignee. Assignee should be selected when the task is created. The reporter of the tasks is the user that is created the task. Users can also create sub-task which is related with any of the existing tasks. The only difference between task and sub-task is that sub-tasks has a root task. Users can log work under any task and indicate how much time they spent and the description of the work.

There are three roles in a single project. Project manager, watcher and team member. Project manager is the user which is the created the project by default. Project managers can create tasks and assign it to one of the team members. Project managers also can view the reports about the project, tasks and team members. Team members can create tasks too but they can assign this task to only themselves. They can log work to the tasks. Users whose have watcher role in the project can only monitor the project. They can not do any operation.

* 1. **Objectives and Success Criteria of the Project**

The objective of the task management system Tasky is providing a strong platform for managing projects and tasks with a web site and mobile application. By developing this system, we aim to increasing the efficiency of the projects, making easier managing and working on the projects and giving the information about the what has been done for the projects and tasks.

The main success criteria is how many projects in the system are successful? Like we mentioned previously we aim to increasing efficiency of the projects. Therefore, if the projects in the system is successful the project is also successful. Also, number of the projects and tasks in the system and the daily usage of the web site and mobile application are important success criteria for us.

* 1. **Definitions, Acronyms, and Abbreviations**

**Tasky:** Name of the proposed task management system.

* 1. **Overview**

This document contains Introduction, Current System, Proposed System, Glossary and References sections.

In the first part of the introduction, we define the purpose of our system. Then in the section 1.2, we explain the actors and functionalities of the system. In the section 1.3 we talk about the objectives and success criteria of the project. We define some terms that is used in this document in the section 1.4. And for the final part of the introduction, we explained how this document is designed.

In the second part we mention some other systems that is using for task management currently.

In the Proposed System section, we explained our system in detail. First we explain the functional and nonfunctional requirements of the project. In the functional requirements section, we explain all the functions that can be done in the system and which actors will be able to use these functionalities. In the nonfunctional requirements we explained the functionalities that is not directly related with the project.

In the System Models section, we describe the scenarios and use cases of the system in order to indicate the interactions between the actors and system. In these sections we clearly explain how the functions of the system should be done. In the Object Model section, we create a UML class diagram in order to define all the classes and their attributes and operations. We create our dynamic model which consist of sequence and state diagram in the Dynamic Model section. In the final section of the System Models, we design the user interfaces of the system.

In the final sections of the RAD document, first we indicate the terminology of the system in the Glossary. Then in the References section, we list our references.

1. **Current System**

**BAŞKASININ TASKINI UPDATE EDEBİLİYORUM**

If the new system **will replace an existing system**, this section describes the functionality and the problems of the current system. For example, assume that you are planning to implement an e-commerce web app. Obviously, there are existing systems like hepsiburada etc. Describe existing systems’ functionality, problems.

If you will work on a scientific problem, present a brief literature survey here, only about the problem.

If you answer no for both cases, remove this part.

1. **Proposed System**

In this project, a task management system which consists of a web and mobile application named Tasky is designed.

* 1. **Overview**

Tasky is an online task management system. The main functions of the system is creating projects and tasks. There are three different roles within each project which defines the accessibility to the system’s functionalities for the users.

First role is the project manager. Project managers are the users who created the project. Project managers can invite users to the project. They can create tasks and sub-tasks. Then they can assign these tasks to the any participant of the project. If a project manager work on a task, he/she can log work under this task. Project managers can view reports about the project, tasks and participants to see the how is project going on.

Second role is team member. Team members are the users who work on tasks within the project. They can log work under the tasks. They can also create task and sub-tasks. But they can assign these tasks to only themselves.

Third role is the watcher. Watcher is the who has no permission to do any operation within the project. They can only monitor the project.

* 1. **Functional Requirements**

Users can register to the system by filling a registration form. They can login to the system with e-mail and password information. They can also logout. After they logged in to the system they have a profile created by their registration credentials. They can view and edit their profile. Users also can manage their preferences which includes turn on/off notifications and change password functions. These functions are available in both platforms.

Users can create and manage projects via Tasky web application. While creating a project they can invite participants by indicating their e-mails. If there is already a user registered with the given e-mail, the system sends an invitation e-mail and mobile notification; otherwise, only an e-mail sent. Invited users can accept or decline the invitation. Creator of the project becomes the project manager. Project managers can assign a role to the project participants. Project managers can update the project by changing the status, name or description of the project. Project managers can add or remove participants to/from projects. Project managers can also delete the project.

Participants of the project can view project detail and project board in both platforms. Project detail contains name, description, participants and documents of the project. Project board contains tasks grouped by their status (To-do, active, resolved and closed).

Project managers and team members can create tasks under their projects via only Tasky web application. These tasks can have sub-tasks. The only difference between the task and sub-task is that sub-tasks have a root task.

These tasks can be assigned to a team member. Project managers can assign a task to anyone in the project except watchers. Team members can only assign tasks to themselves. Project managers and assignee of the task can update the task’s status, description, files. All participants of the project can view the detail of tasks. Project managers and team members can log work to the tasks. A task can be deleted by only a project manager. These functions are available in both platforms.

Project participants can view the task list of the project via both platforms. In this page/screen they can filter or sort the tasks. Users also can view the last activities of their projects.

Project managers can view the report of their projects via only web application. In the project report they can see the stats of tasks or team members.

Project managers will receive e-mail and mobile notifications when:

* A user accepts or declines their invitation.
* A task is updated.
* A work is logged into a task they created.
* A team member created a new task.
* A participant left the project.

Team members will receive e-mail and mobile notifications when:

* A task is assigned to them.
* A work is logged into a task they created.
* A task that is assigned to them is updated or deleted.
* They are removed from a project.
  1. **Nonfunctional Requirements**
     1. **Usability**

Our web and mobile application should be designed according to the UI/UX standards. Operations in applications should be done in minimum number of clicks.

* + 1. **Reliability**

The system should run 24/7. There should be validations in the form inputs in order to avoid crashes and errors.

* + 1. **Performance**

The response time of the system should be maximum 30 seconds for heavy functions like preparing a project report, for light functions it should be maximum 10 seconds.

* + 1. **Supportability**

The system should be seperated into packages/components to accomplish easy maintenance.

* + 1. **Implementation**

The mobile application should be implemented with React-native (JavaScript). Web application should be implemented with React (JavaScript). Web API’s that will be used in the mobile and web application should be implemented with .NET Framework (C#). Database should be MSSQL. Visual Studio Code should be used for React and React-native. Visual Studio 2019 should be used for .NET Framework.

* + 1. **Interface**

Our web and mobile application should be designed according to the UI/UX standards.

* + 1. **Packaging**

Our web application does not require any installation. It should be run on any internet browser. Mobile application should be run on Android and iOS platforms by installing the application.

* + 1. **Legal**

Our web application should have necessary certificates. Mobile application should be signed with a bundle key for android. For iOS, it should be signed with a provisioning profile.

* 1. **System Models**

Describe the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

* + 1. **Scenarios**

A scenario is an instance of a use case.

* + 1. **Use case model**

A use case is a generalization of a number of scenarios. Therefore, the number of scenarios must be equal to or greater than the number of use cases.

* + 1. **Object model**

The analysis object model, depicted with UML class diagrams, includes classes, attributes, and operations. The analysis object model is a visual dictionary of the main concepts visible to the user.

* + 1. **Dynamic model**

The dynamic model is depicted with sequence diagrams and with state machines. Sequence diagrams represent the interactions among a set of objects during a single use case. State machines represent the behavior of a single object (or a group of very tightly coupled objects). The dynamic model serves to assign responsibilities to individual classes and, in the process, to identify new classes, associations, and attributes to be added to the analysis object model.

When working with either the analysis object model or the dynamic model, it is essential to remember that these models **represent user-level concepts, not actual software classes or components.**

* + 1. **User interface—navigational paths and screen mock-ups**

1. **Glossary**

To establish a clear terminology, developers identify the **participating objects** for each use case. Developers should identify, name, and describe them unambiguously and collate them into a glossary.

1. **References**

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.