**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 have 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 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**

There are a lot of task management system in use. Jira, Azure Board and Trello are the most popular task management systems. In these systems basically there are projects, tasks and work logs. Monitoring of the projects are handling with these concepts. There are members within a project. Project members logs work under tasks and update the status of the task. By that way, they can track their tasks and plan their works.

In our system the concept is the same. But our main goal is making processes easier and faster than current systems. We aim to ensure simpler and more undersantable interface. Also, with the mobile application we aim to make accessible our system from everywhere.

1. **Proposed System**

In this project, a task management system which consists of a web and mobile application named Tasky is proposed. This system is designed to make easier and more efficient all the processes that can be done in the project by any member of the project team. For project managers, managing and controlling the project, tasks and team members will become easier. Project managers will be able to see the answers of most of the questions about the project such as:

* Which project member is working on which task?
* How much time is spent for a specific task or project?
* How many hours do project members work for the project in a day?

For team members who working on a task of the project; following the tasks that he/she will do, logging work under the tasks and keeping themselves up to date about the other tasks in the project will be easier.

* 1. **Overview**

The main functions of the Tasky are 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**
* Users should be able to complete their transactions with maximum 5 clicks.
* The components of web and mobile application such as titles, button names, section names etc. should be descriptive and understandable for people who knows English.
  + 1. **Reliability**
* The system should be available 100% of time.
* The system should validate the data that will be stored in the database and if there are errors, users should be notified by the system.
  + 1. **Performance**
* The response time of the system should be maximum 30 seconds for heavy functions like preparing a project report, for other functions it should be maximum 10 seconds.
  + 1. **Supportability**
* The system should be separated 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**
* Mobile application should be signed with a signing key for android. For iOS, it should be signed with a certificate.
  1. **System Models**
     1. **Scenarios**

**Scenario Name:** Login (Web + Mobil)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1. Ali clicks on the Login button on the main page/screen of the Tasky web/mobile application.
2. Ali enters his email and password to login form and clicks to login button.
3. Given login credentials are checked and if they match Ali is redirected to his homepage, otherwise Ali gets an error feedback that declares credentials are wrong.

**Scenario Name:** Register (Web + Mobil)

**Participating Actors:** Berke: Guest

**Flow of Events:**

1. Berke clicks on the Register button on the main page/screen of the Tasky web/mobile application.
2. Berke enters asked information to the registration form.
3. Given credentials are checked and if they are valid an activation email is sent to Berke’s email.
4. Berke clicks on the URL that is sent to his email and activates his account.
5. Registration is completed.

**Scenario Name:** Logout (Web + Mobil)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1. Ali clicks on the logout button which is visible on every page/screen of Tasky web/mobil application.
2. Ali is logged out from the Tasky application.

**Scenario Name:** View Profile (Mobile)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1- In the main screen of Tasky mobile application, Ali clicks on the Profile button from the drawer menu.

2- System redirects Ali to profile screen.

**Scenario Name:** View Profile (Web)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1- Ali clicks on his profile image on top right corner of the Tasky web application.

2- Ali clicks on the Profile button on the opened menu.

3- System redirects Ali to profile page.

**Scenario Name:** Edit Profile (Mobile)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1. Ali clicks on the Profile button from the drawer menu in the main screen.
2. System redirects Ali to profile screen.
3. Ali clicks on the edit icon on the top right corner of the screen.
4. System opens a form that is filled with Ali’s information.
5. Ali edits the information that he wants to change.
6. Ali clicks on the save button.

**Scenario Name:** Edit Profile (Web)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1. Ali clicks on his profile image on top right corner of the Tasky web application.
2. System opens a menu.
3. Ali clicks on the Profile button on the menu.
4. System redirects Ali to profile page. Then Ali clicks on the edit button.
5. System opens a form that is filled with Ali’s information.
6. Ali edits the information that he wants to change.
7. Ali clicks on the save button.

**Scenario Name:** Manage User Preferences (Mobile + Web)

**Participating** **Actors**: Ali: Registered User

**Flow of Events:**

1. Ali clicks on the Preferences button from the drawer/side bar menu in the main screen/page of Tasky mobile/web application.
2. System opens the preferences form.
3. Ali edits the preferences that he wants to change.
4. Ali clicks on the save button.

**Scenario Name:** Create Project (Web)

**Participating Actors:** Ali: Registered User

**Flow of Events:**

1. Ali clicks on the Create Project button on the projects page of the Tasky web application.
2. Ali fills the Project Creation form and clicks to Create button.
3. Project is created and if Ali added any participants, an invitation email sent to their emails.

**Scenario Name:** Update Project (Web)

**Participating** **Actors**: Sude: Project Manager

**Flow of Events:**

1. Sude clicks on the Detail tab of the project that she is the manager of.
2. Sude clicks on the Edit button and changes the field(s) that she wants to update.
3. Sude clicks Save button and project is updated.

**Scenario Name:** Delete Project (Web)

**Participating** **Actors**: Sude: Project Manager

**Flow of Events:**

1. Sude clicks on the Detail tab of the project that she is the manager of.
2. Sude clicks on the Delete button.
3. Sude enters her password to confirm deletion.
4. Project is deleted.

**Scenario Name:** Invite Participant to Project (Web)

**Participating** **Actors**: Ayşe: Project Manager

**Flow of Events:**

1. Ayşe clicks on the projects button in the side bar menu and opens the list of the projects. Then she navigates to the project detail by clicking on the project name.
2. Ayşe clicks on the add participant button under the project participants section in the project detail.

3- Ayşe writes the e-mail of the person that she wants to invite and clicks on the invite

button.

4- System sends an e-mail and mobile notification to the invited users.

**Scenario Name:** Remove Participant from Project (Web)

**Participating** **Actors**: Ayşe: Project Manager

**Flow of Events:**

1. Ayşe clicks on the projects button in the side bar menu and opens the list of the projects.
2. She navigates to the project detail by clicking on the project name.
3. Ayşe clicks on the remove button which is next to the name of the participant that she wants to remove.
4. System opens a warning box to ask she is sure to remove that participant.

5- Ayşe confirms that she wants to remove selected user from the project by clicking the remove button in the warning box.

**Scenario Name:** Accept/Decline Invitation (For Registered User)

**Participating** **Actors**: Ahmet: Registered User

**Flow of Events:**

1. Ahmet receives an e-mail and mobile notification that tells he is invited to a project.
2. Ahmet clicks on the url in the e-mail or mobile notification.

3- Ahmet displays the information of the project that he is invited. He accepts or declines the invitation by clicking the accept or decline button.

4- Manager of the project gets a notification about the Fatma’s response.

**Scenario Name:** Accept/Decline Invitation (For Guest)

**Participating** **Actors**: Fatma: Guest

**Flow of Events:**

1. Fatma receives an e-mail that tells she is invited to a project.
2. Fatma clicks on the url in the e-mail.
3. System opens the registration form.

4- Fatma fills that form and clicks on the sign-up button.

5- System opens the project information page.

6- Fatma accepts or declines the invitation by clicking the accept or decline button.

7- Manager of the project gets a notification about the Fatma’s response.

**Scenario Name:** View Project Detail (Web + Mobil)

**Participating** **Actors**: Kemal: Project Participant

**Flow of Events:**

1. Kemal clicks on the project’s name that he wants view details in the projects page/screen.

2- Kemal clicks on the Details tab on the project page/screen.

**Scenario Name:** View Project Report (Web)

**Participating** **Actors**: Sude: Project Manager

**Flow of Events:**

1. Sude clicks on the project’s name that she wants view report of, in the projects page.
2. Sude clicks on the Report tab on the project page.

**Scenario Name:** View Project Board (Web + Mobil)

**Participating** **Actors**: Ahmet: Project Participant

**Flow of Events:**

1. Ahmet clicks on the project’s name on projects tab.
2. Ahmet clicks on the Board tab in project page/screen.

**Scenario Name:** Create Task

**Participating** **Actors**: Ayşe: Project Manager, Kemal: Team Member

**Flow of Events:**

1. In the main page of Tasky web application, Ayşe/Kemal clicks on the name of the project on the side bar menu and navigates to project detail.
2. Ayşe/Kemal clicks on the create new task button.

3- Ayşe/Kemal fills the new task form and clicks on the create button.

4- If Ayşe selects an assignee for the created task, system sends a notification to the assignee about this task is assigned to him/her.

**Scenario Name:** Create Sub-task

**Participating** **Actors**: Ayşe: Project Manager, Kemal: Team Member

**Flow of Events:**

1. In the main page of Tasky web application, Ayşe/Kemal clicks on the name of the project on the side bar menu and navigates to project detail.
2. Ayşe/Kemal displays the task list and clicks on the name of the task. Then he/she navigates to task detail
3. Ayşe/Kemal clicks on the create sub-task button.

4- Ayşe/Kemal fills the new sub-task form and clicks on the create button.

5- If Ayşe selects an assignee for the created task, that assignee gets a notification about this task is assigned to him/her.

**Scenario Name:** View Task List (Web + Mobil)

**Participating** **Actors**: Ahmet: Project Participant

**Flow of Events:**

1. Ahmet clicks on the project’s name on projects tab.
2. Ahmet clicks on the Tasks tab in project page/screen.

**Scenario Name:** Filter Task List (Web + Mobil)

**Participating** **Actors**: Ahmet: Project Participant

**Flow of Events:**

1. Ahmet clicks on the project’s name on projects tab.
2. Ahmet clicks on the Tasks tab in project page/screen.
3. Ahmet selects which filter(s) he wants to use for filtering and clicks them.
4. Task list is filtered according to selected options.

**Scenario Name:** Sort Task List

**Participating** **Actors**: Ahmet: Project Participant

**Flow of Events:**

1. Ahmet clicks on the project’s name on projects tab.
2. Ahmet clicks on the Tasks tab in project page/screen.
3. Ahmet clicks on the column that he wants to order by.
4. Task list is sorted according to clicked column.

**Scenario Name:** Delete Task

**Participating** **Actors**: Ayşe: Project Manager, Kemal: Team Member, Oğuz: Assignee of the Task

**Flow of Events:**

1. In the main page of Tasky web application, Ayşe clicks on the name of the project on the side bar menu and navigates to project detail.
2. Ayşe displays the task list and clicks on the name of the task.
3. System opens the task detail.
4. Ayşe clicks on the delete task button.

5- Ayşe confirms that she wants to delete this task. Then system deletes the task.

6- Oğuz gets a notification about the task is deleted.

**Scenario Name:** View Task Detail (Web)

**Participating** **Actors**: Ayşe: Project Manager, Kemal: Team Member, Zeynep: Watcher

**Flow of Events:**

1. In the main page of Tasky web application, Ayşe/Kemal/Zeynep clicks on the name of the project on the side bar menu and navigates to project detail.
2. Ayşe/Kemal/Zeynep displays the task list and clicks on the name of the task. Then he/she navigates to task detail

**Scenario Name:** View Task Detail (Mobile)

**Participating** **Actors**: Ayşe: Project Manager, Kemal: Team Member, Zeynep: Watcher

**Flow of Events:**

1. In the main screen of Tasky mobile application, Ayşe/Kemal/Zeynep clicks on the Projects tab from the bottom tab menu.
2. System opens the projects list.
3. He/she clicks on the project name and navigates to project detail.
4. Ayşe/Kemal/Zeynep displays the task list and clicks on the name of the task. Then he/she navigates to task detail

**Scenario Name:** Log Work (Mobile)

**Participating** **Actors**: Kemal: Team Member, Oğuz: Reporter of the Task

**Flow of Events:**

1. In the Tasky mobile application, Kemal navigates to task detail from the main screen or task list screen.
2. He clicks on the Log Work button.
3. System opens a work log form.
4. He fills the form and clicks on submit button.
5. Oğuz gets a notification about the logged work.

**Scenario Name:** Log Work (Web)

**Participating** **Actors**: Kemal: Team Member, Oğuz: Reporter of the Task

**Flow of Events:**

1. In the Tasky web application, Kemal navigates to task detail from the main page.
2. He clicks on the Log Work button.
3. System opens a work log form.
4. He fills the form and clicks on submit button.
5. Oğuz gets a notification about the logged work.

**Scenario Name:** Update Task

**Participating** **Actors**: Kemal: Team Member, Ayşe: Project Manager

**Flow of Events:**

1. In the Tasky web application, Ayşe/Kemal navigates to task detail from the main screen or task list screen.
2. He/She clicks on the Edit button.
3. System opens the Task form.
4. He/She edits the information that he/she wants to change. Then he/she clicks on the save button.

**Scenario Name:** Update Task Status (Web)

**Participating** **Actors**: Kemal: Team Member, Ayşe: Project Manager

**Flow of Events:**

1. In the Tasky web application, Ayşe/Kemal navigates to project board from the project detail page.
2. He/She drags the task and drops it to the status column that he/she wants.

**Scenario Name:** Update Task Status (Mobile)

**Participating** **Actors**: Kemal: Team Member, Ayşe: Project Manager

**Flow of Events:**

1. In the Tasky mobile application, Ayşe/Kemal navigates to task detail from the main or task list screen.
2. He/She clicks on the status text and selects the new status of the task.

**Scenario Name:** View Last Activities (Web)

**Participating** **Actors**: Kemal: Team Member, Ayşe: Project Manager, Zeynep: Watcher

**Flow of Events:**

1. In the Tasky web application, Ayşe/Kemal/Zeynep logs into the system.
2. System opens the last activities list.

**Scenario Name:** View Last Activities (Mobile)

**Participating** **Actors**: Kemal: Team Member, Ayşe: Project Manager, Zeynep: Watcher

**Flow of Events:**

1. In the Tasky mobile application, Ayşe/Kemal/Zeynep clicks on the Activities tab.
2. System opens the last activities list.
   * 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.