**Planning**

[1. Context and problematic 2](#_Toc523840353)

[2. Analysis of the existing 2](#_Toc523840354)

[**1.1.** **TRELLO** 2](#_Toc523840355)

[**1.2.** **WRIKE** 3](#_Toc523840356)

[**1.3.** **ASANA** 3](#_Toc523840357)

[3. Project objective 3](#_Toc523840358)

[4. Functional requirements 4](#_Toc523840359)

[5. Used technologies and framework 5](#_Toc523840360)

[**1.1.** **Angular** 5](#_Toc523840361)

[**1.2.** **Node.js** 5](#_Toc523840362)

[**1.3.** **MongoDB and Azure Cosmos DB** 5](#_Toc523840363)

[6. Model 6](#_Toc523840364)

# **Context and problematic**

Managing a project can determine the success of that project. Only 2.5% of the companies successfully complete their projects There are many reasons why projects (both simple and complex) fail; And further. That many of them fail for the same reasons.

* Unrealistic deadlines
* Improper communication
* Geographically dispersed teams
* Lack of control
* Wrong task for the wrong member of the team
* Scope creep
* Wasting time on an already solved issue
* Not using a project management software or tool

Project management isn’t what it seems to be - it’s much more than that. It’s a skill that takes time, patience and perseverance. It requires you to constantly analyze setbacks and failures and implement the lessons learned from them. Which will consume a lot of energy, time and money.

# **Analysis of the existing**

In order to determine the requirements of our project we decided to go through multiple existing software concerning project management. TRELLO, WRIKE and ASANA will be presented briefly in the next section.

## **TRELLO**



Web-based project management and collaboration tool-based Kanban-like lists. TRELLO provide the user with different functions:

* create and manage boards
* divide the tasks between members manually
* divide a task into a checklist
* assign deadline for every task
* provide a task with details such as a file or a link
* get notified regarding any changes in the board or task

## **WRIKE**



WRIKE also a project management solution that enable uses to track projects, monitor, prioritize tasks as well as collaborate with the people in multifunctional teams to get work done. It helps the users to:

* Builds a path
* Set a timeline
* Prioritize and visualize
* Check the outcome
* Analyze results

## **ASANA**



ASANA is an application that helps teams to organize, track and manage their projects.

Key features:

* Update and add tasks
* Communicate with teammates
* Track and organize the work
* Get notification or reminders

# **Project objective**

We aim to create a project management tool that provides our clients with a safe space to communicate, share solved issues and suggest additions to a project. In addition to that. This tool will help the user to divide the tasks on the team members and make it easier to determine the deadlines of every task in the project.

# **Functional requirements**

* Manage teams
* Create a team
* Modify a team
* Add members
* Remove a member
* Modify the name of the team
* Delete a team
* Manage boards (projects)
* Create a new board: after creating a project with all the members, cards and tasks, the tool can give an approximation of the duration of the project and assign to every member of the team a specific task
* Modify a board
* Rename a board
* Modify the deadline of the project
* Change the team in charge
* Manage cards
* Add a card
* Modify card
* Rename
* Observe or stop observing a card
* Delete card: the card can not be deleted without the permission of the project manager
* Manage tasks
* Add a task to a specific card
* Modify task
* Rename
* Change members
* Change deadline
* Modify description
* Delete task: a task can not be deleted without the permission of the team manager or the members which are responsible for it
* Move a task from a card to another
* Assign a task to one of the team members: this function can either be done manually or automatically
* Find similar tasks in another board: this function is automatically done the moment the task is created
* Manage checklist
* Voting
* Communicate with other clients or team members
* Gantt diagram created automatically
* Notifications send and received

# **Used technologies and framework**

## **Angular**

**Angular** is a JavaScript framework for building client applications in HTML and TypeScript. it is primarily sustained by Google together with an extended community of people and companies, to approach many of the challenges faced when developing single page, cross platform, performant applications. It is fully extensible and works well with other libraries.

Angular is a great tool that will:

* Enable the developer to create software quicker and with less effort
* Result in a more maintainable software
* Encourage good programming practices and design patterns like MVC
* Make collaborate easier with other people
* Address problems that may arise in most software architecture such as Dependency Injection, DRY (Don’t Repeat Yourself), etc.

## **Node.js**

For a runtime environment application **Node.js** is one of the best choices. It allows the programmer to write server-side applications in JavaScript. Thanks to its unique I/O model, it stands out at the sort of real-time situations we are increasingly demanding of our servers.  It’s also lightweight, efficient, and its ability to use JavaScript on both frontend and backend opens new avenues for development.

## **MongoDB and Azure Cosmos DB**

**MongoDB** is an open-source, non-relational database management system (DBMS) that uses a document-oriented database model which support various forms of data. It offers a rich set of features and functionality far beyond those offered by NoSQL data stores.

**Azure Cosmos DB** is a Microsoft’s globally distributed, multi-model database with a click of a button. Azure Cosmos DB is a good choice for new web, mobile, gaming, and IoT applications where automatic scale, predictable performance, fast order of millisecond response times, and the ability to query over schema-free data is important.

Azure Cosmos DB databases can be used as the data store for apps written for [MongoDB](https://docs.mongodb.com/manual/introduction/). your application written for MongoDB can now communicate with Azure Cosmos DB and use Azure Cosmos DB databases instead of MongoDB databases. Using this functionality makes it easier to build and run MongoDB globally distributed database applications in the Azure cloud with Azure Cosmos DB and its [comprehensive industry-leading SLAs](https://azure.microsoft.com/support/legal/sla/cosmos-db), while continuing to use familiar skills and tools for MongoDB.

# **Model**

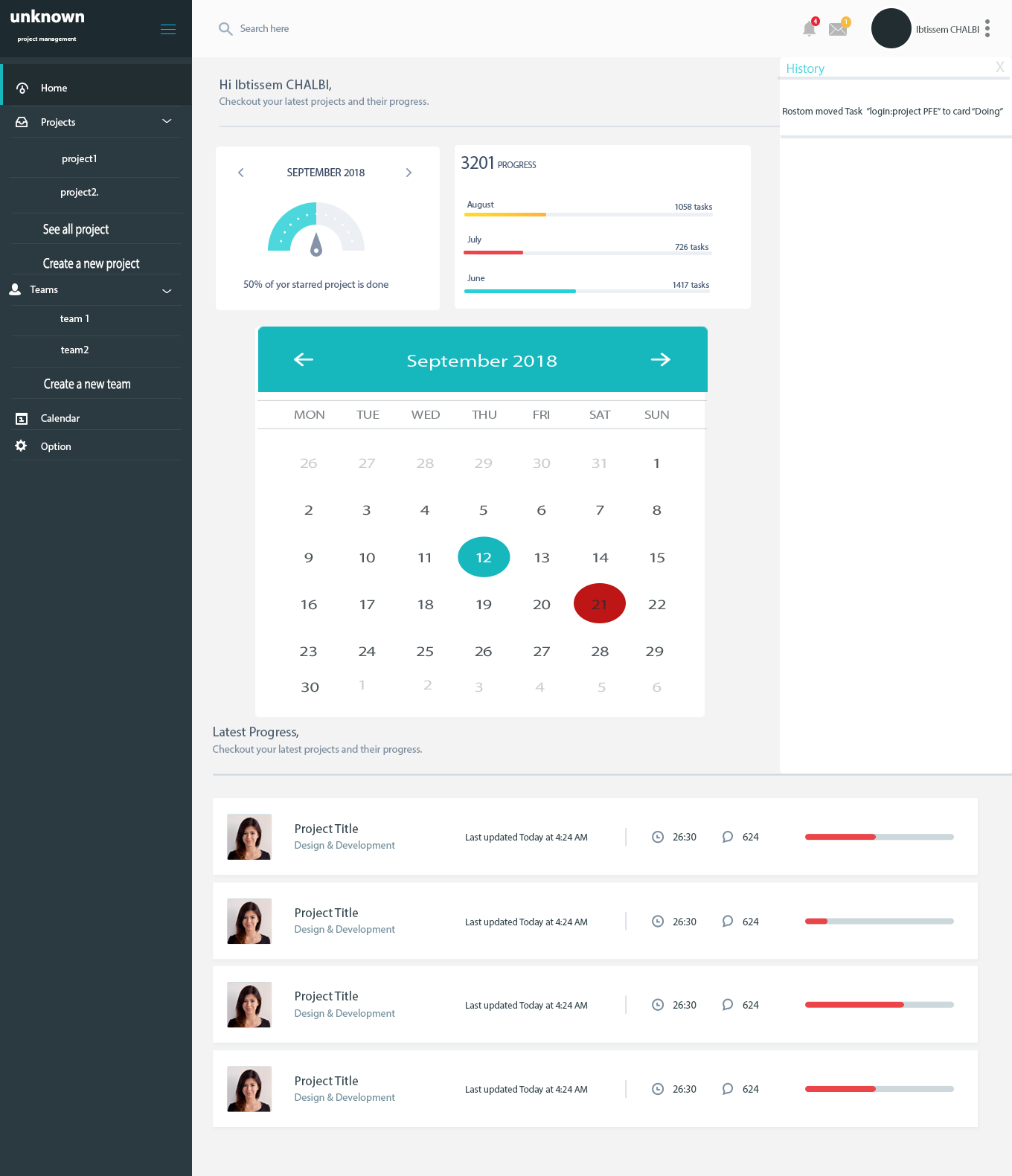


Figure 1: home page of the application

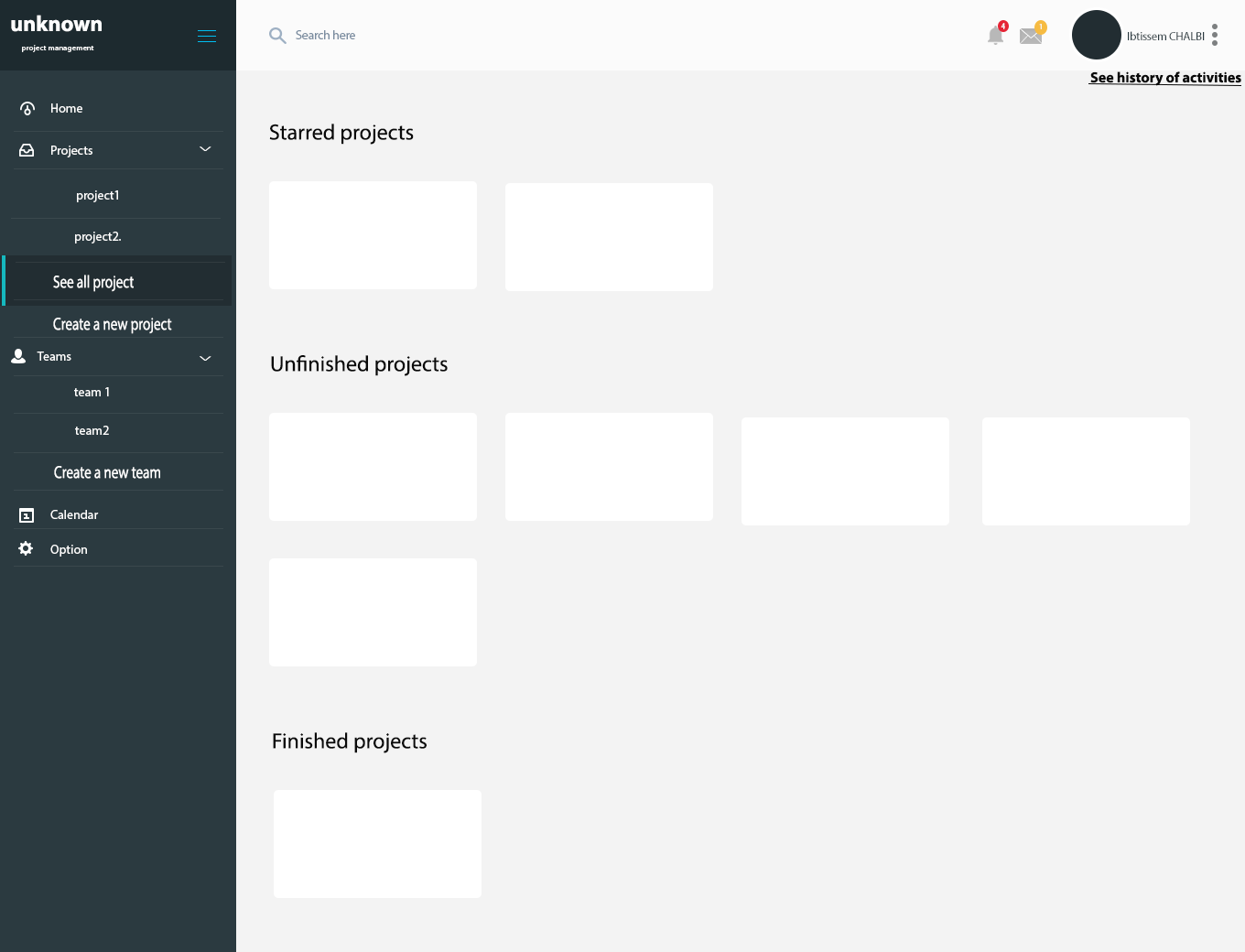


Figure 2: projects list

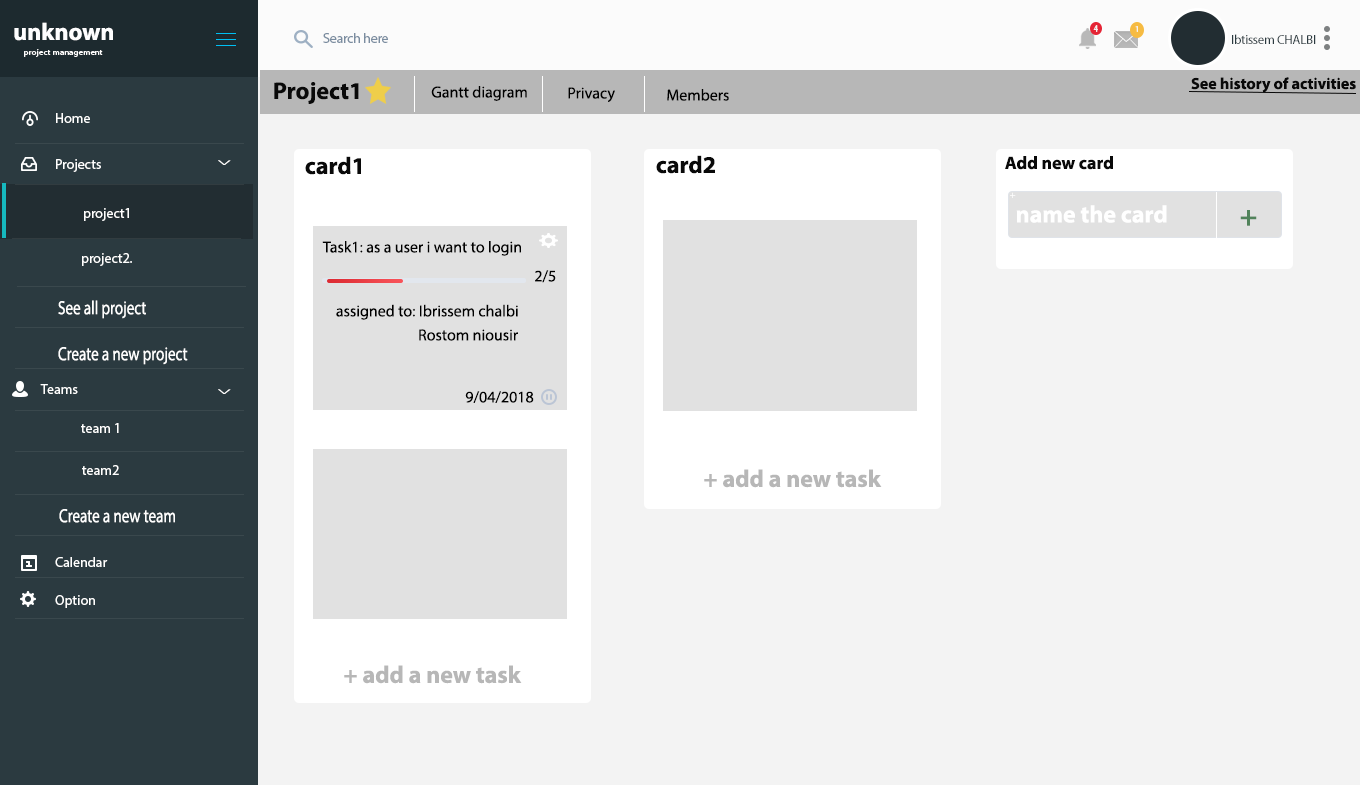


Figure 3: show the board of a project

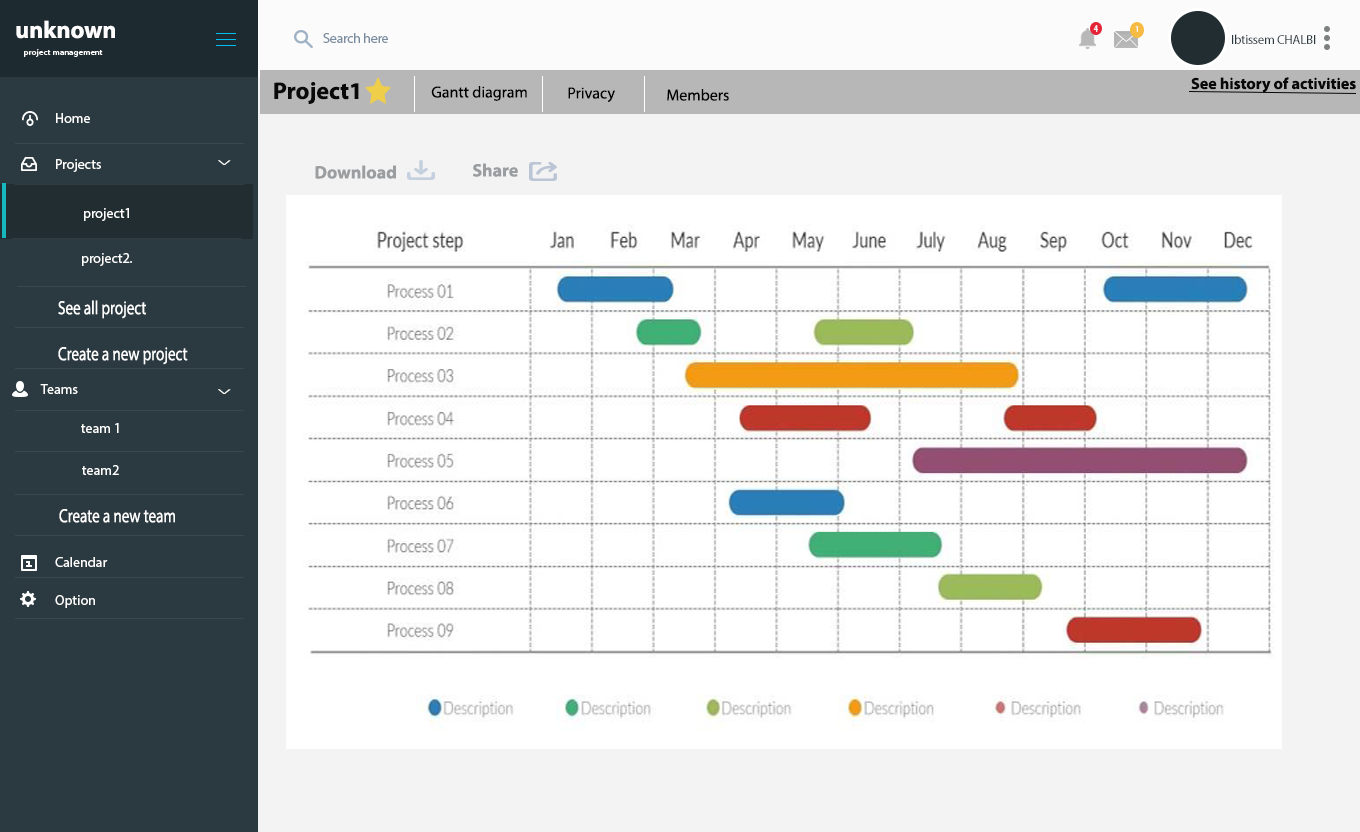


Figure 4: Gant diagram for a specific project