ARSE: Agile Responsive Simple Environment

Team Echo: Project Report for the course Internet Praktikum TK in WS 2015/16

Elmi Faisal Ali
Enkhtuul Gankhuyag
Floriment Klinaku
Masaud Yakubu Alhassan
Omar Antonio Erminy Ugueto
Satia Herfert

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Contents

1 Introduction

During the last decade, agile methodologies have grown in popularity. Big and small companies are dropping old-fashioned development approaches (like the waterfall model) and migrating to new, different techniques, such as Kanban or Scrum. Many of these new agile methodologies depend on pen and paper to keep track of the activities involved in a project. Although this might be sufficient for some teams, things become complicated when the number of people participating in a project increases and it becomes almost unfeasible when the team is distributed around the world. Supporting the development process with a reliable management tool has become an enabling factor to overcome the obstacles in large scale projects.

ARSE offers the possibility to manage agile projects using Scrum. Users are able to create projects, define their product backlogs, schedule sprints and interact with the sprint board. Furthermore, product owners can remove and include participants to the projects they have created; participants can be assigned to product backlog items.

The way a user interacts with an applications has a high impact in the adoption of a new tool. For this reason, usability was an important factor for us when designing the application. We introduce a tool with an interactive and responsive user experience, for example, the user updates the status of a story via a drag and drop interface.

The application was built on top of the MEAN stack and it was developed during the course of the TK Internet Praktikum in the winter semester of 2015/16 at TU Darmstadt. A beta version of ARSE can be found at `http://arsemvp.herokuapp.com/`; in the future we plan to release ARSE as an open-source tool that teams all over the world could use and improve.

The following report describes ARSE in both the technical implementation as well as the use cases of the tool. The next section describes the architecture in technical terms. The user documentation describes the application in terms of the features that are offered to end-users. We conclude with the product backlog items that were included in ARSE as explained in the project specification.

2 Architecture

ARSE is built with the MEAN stack. The MEAN stack consists of **MongoDb**, **Express.js**, **Angular.js** and **Node.js**. Figure 2.1 illustrates the general architecture of an application built on top of the MEAN stack.

Node.js is JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js is used for building server side JavaScript web applications [1].

Express.js is web application framework for Node.js. It represents the backend of the application, handling aspects as API endpoints, controllers and database access via Mongoose (an object modelling mapper for MongoDb) [2].

MongoDb is the database part of the application. It is a document oriented database where data is organized in JSON-like documents. MongoDb is particularly efficient when dealing with unstructured data [3].

Angular.js is a web application framework that provides client-side model-view-controller(MVC). Angular.js is the front end part of the application [4].

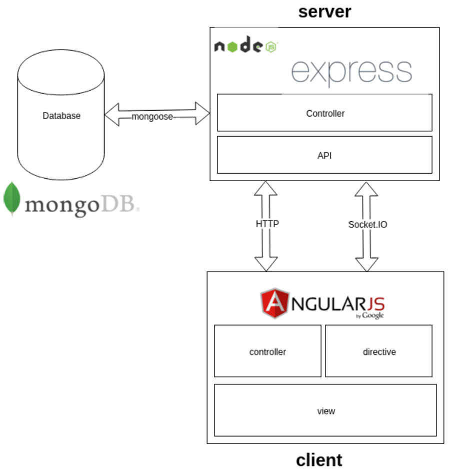


Figure 2.1: The architecture of ARSE

Figure 2.2 shows a class diagram of ARSE that explains the main entities used in the application as well as their relationships. In the next section we explain the main responsibilities of each class.

Project: It represents the most important class in the application. The whole functionality of ARSE revolves around the creation and management of projects. A project has a name and a description that identify it. It contains a list of participants that can have access to the project's information; participants can execute different actions depending on their role in the project. Moreover, each project has a backlog that consists of a list of user stories. Using an offset this ordered backlog is split into the Sprint backlog and the remaining stories. The backlog defines all the requirements of the product or service that is being built. During its lifetime, a project will have several sprints; a list of all past sprints is maintained as well as a reference to the current active sprint if any. Finally, each project has a private chat where all the participants can communicate.

Additionally, each project can maintain a list of additional story types and statuses that will be valid only for that project. These new configuration values can be added by the product owner of the project.

Participant: It represents an abstraction of a user inside the project. The product owner of a project can add or remove participants to a project with a particular role. The roles available are "Product Owner" and "Developer". A reference to the application user is kept in order to access identification attributes such as name and email.

User: It represents a user of the application. A new entry is created every time someone registers in ARSE. This class was modeled using the authentication framework of the "AngularJS Full-stack Generator" [5] and it serves as our main authentication and authorization mechanisms. It provides basic security features, as password encryption and session persistence via cookies.

Story: It represents the user stories that belong to a project. User stories conform the building blocks of Scrum as they allow the product owner to guide the development process by prioritizing the features to be developed. A Story contains a brief summary and a detailed description, as well as the estimated story points. During a sprint, stories undergo multiple status updates that indicate the development progress in that feature. A set of statuses is defined by default, however, the product owner can customize it to the team's needs by adding or removing statuses. This augmented list of statuses can only be used when the story is in the status "In progress". A similar situation occurs with the story types, although they can be set at any point. Moreover, a story can contain a list of tasks that breaks down the user story into particular activities.

Task: It represent a detailed activity that is part of a user story. A tasks has no meaning by itself as it has to be associated with a user story. It contains a summary and a description of the activity to be done, as well as a status that describes its progress. However, the extended list of statues available in the project does not apply to tasks.

Sprint: It represents the execution and outcome of a sprint in the project. During its lifetime, a project will undergo multiple sprints. A sprint is defined by its starting date, its ending date and the total number of story points burnt. Some of these attributes might be empty if the sprint has not finished.

Chat: It represents the chat messages exchanged by the participants of a project. Every message is stored so it could be browsed again later on. A chat message is defined by the user who posted it, the text content of the message and the date when it was sent.

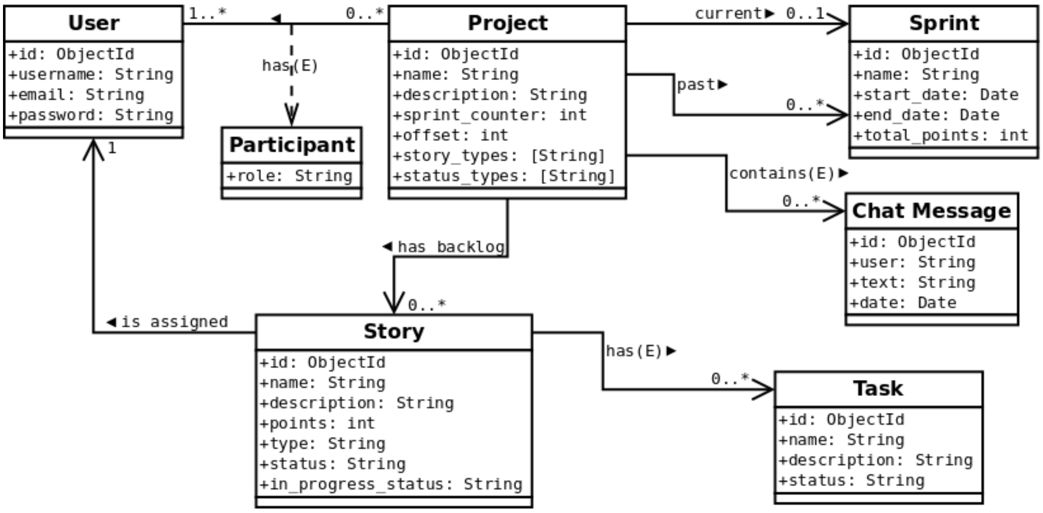


Figure 2.2: Class Diagram. Relationships marked with an "(E)" indicate that the information of the incoming class is embedded in the outgoing class. The remaining relationships are reference-based.

Besides the general explanation provided before, here are some relevant technical aspects of ARSE that influenced the most important design decisions.

- The use of embedded documents inside Mongo collections, for example in the case of tasks being contained inside stories. This allows to keep the information close together to the portions of data that have joint semantic meaning, as well as improving the performance of database queries.
- The so called Express middlewares in endpoint routes that pipeline requests through common actions, for example authentication and authorization. Reusability and maintainability were improved significantly.
- Angular directives also helped with the reusability of front-end code. Modal windows were used in several places throughout the application although the concept of the modal was implemented only once.
- Socket.io allowed the asynchronous communication of events pushed by the server to all the clients. It enabled the inclusion of real-time functionality such as the chat [6].
- Twitter bootstrap and CSS preprocessors provided the means for constructing a platform focused primarily in the user experience both in mobile devices and desktop environments [7].

3 User Documentation

In this chapter we will briefly explain how to use the main features of our application. For each state or thematic group of use cases, we present a section that explains the graphical layout and the possible ways to interact with the tool.

3.1 Profile Management

When first visiting ARSE, you will be asked to login or register. The screen you will see is displayed in figure 3.1. Here you can input the right information to login (1) or click *Register* (2) to switch to a different form where you are asked to register providing your full name, email address and a password. Email addresses serve as usernames and must be unique in the system.

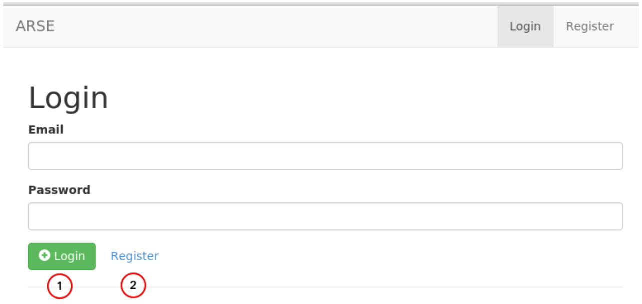
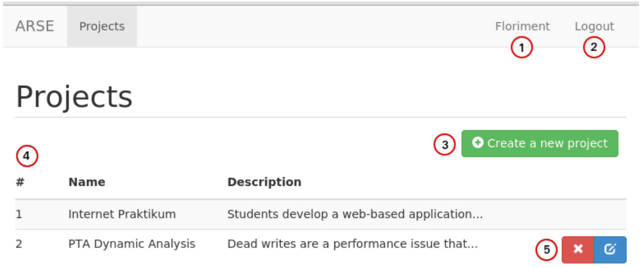


Figure 3.1: The login screen.

3.2 Project Management

After logging in, you will be redirected to a page that shows the projects you are a member of, as depicted in figure 3.2. You can click on your name in the upper right corner of the navigation bar in order to access a page that allows you to edit your profile (1). In that page you can adjust your name and password, but not your email address. The email address cannot be changed after the registration. Changes to your profile always require a password confirmation using your current password. In the upper right corner you will also find a button that logs you out from the system (2).



#	Name	Description
1	Internet Praktikum	Students develop a web-based application...
2	PTA Dynamic Analysis	Dead writes are a performance issue that...

Figure 3.2: The project list screen.

If you just registered in the application, the projects page will be empty. You can now create your own projects or ask colleagues to add you to their projects.

In order to create a project, click the button *Create a new project* (3). This will open a form where you need to fill out name and description of the project. After that, the project will appear in the list (4) and you will automatically become product owner of this project. This gives you privileges that normal developers in a project do not have.

The projects are listed with name and description. If you are product owner of a project, some control buttons will be available (5); you can use them to delete or modify a project.

You can click the name of a project in the list to access the project page. When you click it, you are redirected to a project specific space, which includes a navigation bar to access the different views available for each project. On the left-hand side of figure 3.3, the project navigation bar is displayed (1). On mobile devices, this navigation bar is displayed at the top instead. It includes five icons that trigger different project-related views. Each view is explained in detail in the following subsections.

Product Backlog: A list of all stories of the project can be viewed here. Stories can be added, removed and edited. Furthermore, it is possible to start a new sprint in this view.

Sprint Board: This view can only be accessed if a sprint is currently running. Its purpose is to allow changing the status of stories, managing the tasks of stories, or to close or cancel a sprint.

User management: The participants of the project are displayed in this view. As a product owner, it is also possible to add or remove members as well as to change their roles in the project.

Past Sprints: This view simply lists all finished sprints, with their time-frames and the number of story points that were burnt in the sprint.

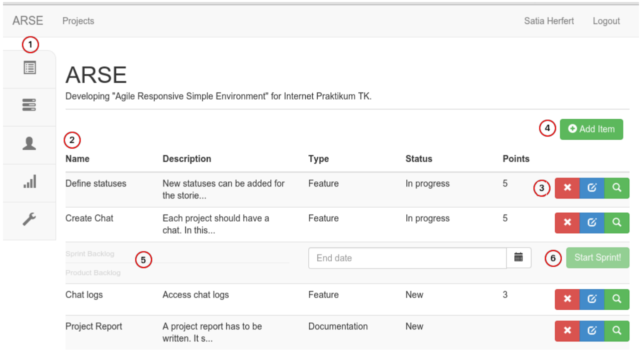
Project configuration: This view is only accessible by product owners; the icon will be hidden for developers. The view allows you to define additional story types and statuses. These can then be used on every story of the project later on.

3.2.1 Product Backlog

This view lists all stories of a project, in priority order (2). An example of how this can look is given in figure 3.3. For each story, name, description, type, status and story points are given. Three buttons follow in each row of the table (3). The first two allow to delete or modify a story; the last displays more details about the story on the left side of the screen.

In the top there is a button that can be used to add a new story (4). The required information is a name and a description, the number of story points, the story type and the assignee. It is possible to leave a story without an assignee. The form shown has the same layout as when editing stories.

To change the order –and thereby the priority– of stories, users can simply drag and drop the rows up or down in the table. There is one special row that is always present: the sprint delimiter (5). In the example screenshot, this is the third row. This delimiter separates the sprint backlog (above) from the product backlog (below). It can be dragged like any other row, effectively changing the scope of a sprint. This will issue a warning, if a sprint is currently active. The sprint backlog is the subset of all stories that are expected to be done by the end of the sprint.



ARSE

Developing "Agile Responsive Simple Environment" for Internet Praktikum TK.

Name	Description	Type	Status	Points
Define statuses	New statuses can be added for the storie...	Feature	In progress	5
Create Chat	Each project should have a chat. In this...	Feature	In progress	5
Sprint Backlog Product Backlog		End date		
Chat logs	Access chat logs	Feature	New	3
Project Report	A project report has to be written. It s...	Documentation	New	

Figure 3.3: The project backlog screen with the project navigation bar on the left-hand side.

Finally, in the row of the sprint delimiter, product owners have the possibility to start a new sprint. In order to that, first the planned end date has to be chosen and afterwards the button *Start Sprint!* can be clicked (6).

3.2.2 Sprint Board

This view becomes available after a sprint has been started. Given the big difference in screen sizes, the view is designed differently for desktop and mobile versions. In the following, first the desktop version is explained with all available features, and then the key differences to the mobile version are pointed out. An example of the desktop version is given in figure 3.4a, and for the mobile version in figure 3.4b.

Desktop version

In the top there are two buttons displayed for product owners. They can be used to cancel or close a sprint respectively (1). Canceling means that all stories are transferred back to product backlog, regardless of their status. When closing a sprint, only the stories not done are transferred back; the finished stories are deleted. Please note that this action is irreversible and data may be lost if clicking this by accident.

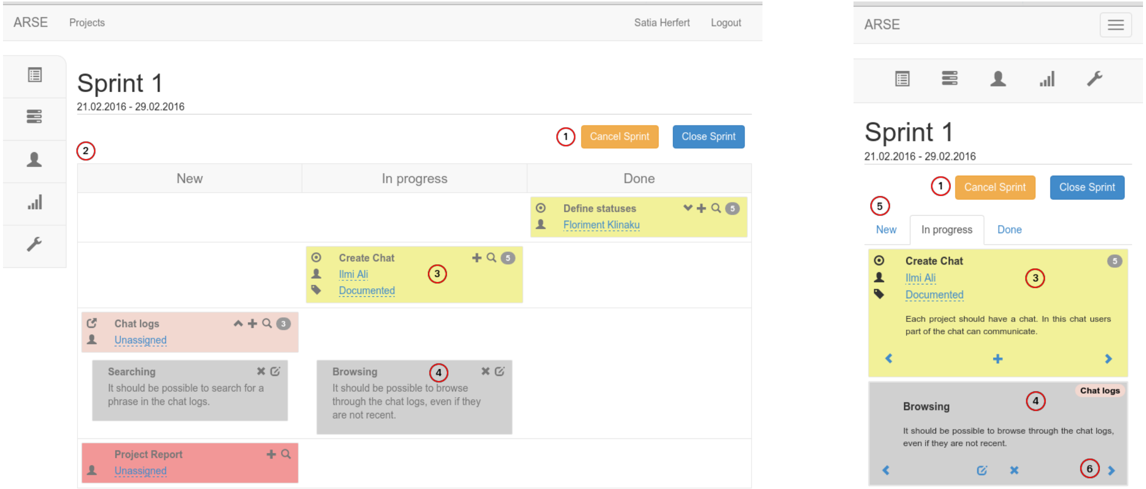
Below that, there is a table that resembles a physical sprint board in agile projects (2). There are three columns, for new, started, and finished stories. Each story has its own row in the table and is displayed in a colored card providing essential information about it (3). The cards can be dragged and dropped into other columns to change the status of the story.

The color of the card and the upper left icon are chosen according to the its type. In addition, you can see the name of the story, and the associated story points. The plus icon allows to add tasks to the story, which are explained below. The magnifying glass shows all details of the story on the right-hand side of the screen, in the same way as in the product backlog view. The person assigned to a story is displayed in a second line on each card, and it can also be changed in-place. Moreover, stories that are in progress (second column), allow to have an additional status assigned. This is a tag that can be used if the three statuses do not suffice. Product owners can configure these tags in the project configuration. This modifiable tag is displayed in a third line, if tags have been defined.

If tasks have been added to a story, they are displayed in gray cards below the story with title and description (4). Just as stories, they can be dragged to other columns to change their statuses. These gray cards also have icons to edit or delete the task. To show or hide the tasks of a story, the arrow pointing down or up on the story card can be clicked.

Mobile version

On the small screens of mobile devices it is not possible to show all three columns simultaneously Therefore, the table has been replaced by a set of three tabs for the respective statuses (5). Now, all information is displayed on the cards, including the description, and no magnifying glass icon is available. To change the status of a story or task, right and left arrow icons have been added (6). Tasks are always displayed in the mobile version and cannot be hidden as in the desktop version. As it is possible that a task is in a different tab as the story it belongs to (for example the task *Browsing* in figure 3.4), the name of the story is always displayed in the upper right corner of a task card. The controls to change assignees or tags, and the buttons to close or cancel a sprint have not been changed.



It should be possible to search for a phrase in the chat logs.

It should be possible to browse through the chat logs, even if they are not recent.

Each project should have a chat. In this chat users part of the chat can communicate.

(a)

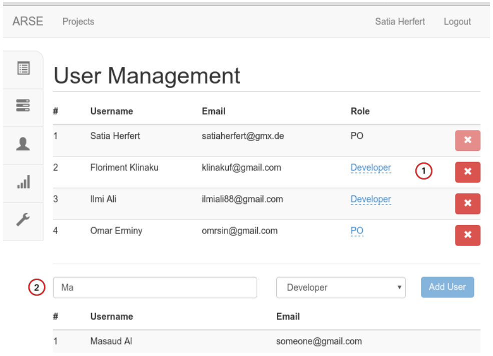
(b)

Figure 3.4: The sprint board in its desktop and mobile versions.

3.2.3 User Management

The user management view, depicted in Figure 3.5, lists all participants of the project. All developers can see the names, email addresses and roles of their project colleagues. As a product owner, it is also possible to change the role of other participants, or to remove users from the project (1).

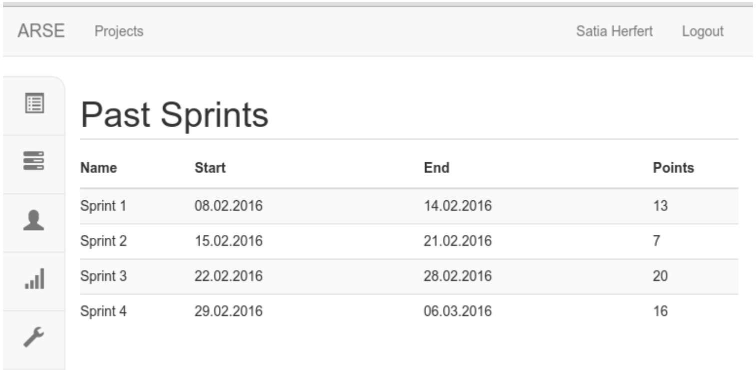
Product owners also see another set of controls below, that allows them to add new participants to the project (2). A text field can be used to search for usernames or email addresses. By clicking the corresponding row in the search results, a user is chosen to be added. Finally, the role of the new user can be chosen between *Developer* and *PO* (product owner), before clicking the *Add user* button.



#	Username	Email	Role
1	Satia Herfert	satiaherfert@gmx.de	PO
2	Floriment Klinaku	klinakuf@gmail.com	Developer
3	Ilmi Ali	ilmiali88@gmail.com	Developer
4	Omar Erminy	omrsin@gmail.com	PO

#	Username	Email
1	Masaud Al	someone@gmail.com

Figure 3.5: The user management of a project.



Name	Start	End	Points
Sprint 1	08.02.2016	14.02.2016	13
Sprint 2	15.02.2016	21.02.2016	7
Sprint 3	22.02.2016	28.02.2016	20
Sprint 4	29.02.2016	06.03.2016	16

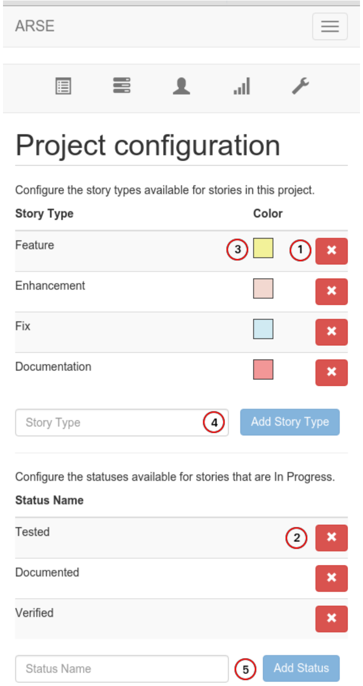
Figure 3.6: The past sprints of a project.

3.2.4 Past Sprints

This view is solely informative and does not contain any controls; an example can be seen in figure 3.6. A table lists all past sprints with their respective start and end dates, and the amount of story points that were burned in the sprint. This is the sum of the story points of stories that were finished in the sprint.

3.2.5 Project Configuration

The project configuration view is only accessible by product owners; it is depicted in 3.7. It has two purposes: managing story types and additional statuses (tags) of stories. For both, a list of the currently defined items is shown, and a button to remove a story type or status (1)(2). The first table also displays the colors of the story types that are used in the sprint board (3). Moreover, there are text fields and buttons to add new types or statuses (4)(5).



Project configuration

Configure the story types available for stories in this project.

Story Type	Color	
Feature	🟨	×
Enhancement	🟥	×
Fix	🟦	×
Documentation	🟥	×

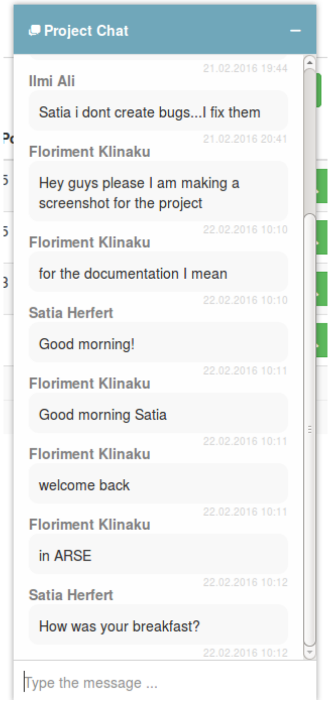
Story Type | Add Story Type

Configure the statuses available for stories that are In Progress.

Status Name	
Tested	×
Documented	×
Verified	×

Status Name | Add Status

Figure 3.7: The project configuration.



21.02.2016 19:44
Ilmi Ali
Satia i dont create bugs...I fix them

21.02.2016 20:41
Floriment Klinaku
Hey guys please I am making a screenshot for the project

22.02.2016 10:10
Floriment Klinaku
for the documentation I mean

22.02.2016 10:10
Satia Herfert
Good morning!

22.02.2016 10:11
Floriment Klinaku
Good morning Satia

22.02.2016 10:11
Floriment Klinaku
welcome back

22.02.2016 10:11
Floriment Klinaku
in ARSE

22.02.2016 10:12
Satia Herfert
How was your breakfast?

22.02.2016 10:12

Type the message ...

Figure 3.8: The Chat Window.

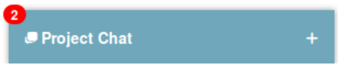


Figure 3.9: The Chat Window minimized.

3.2.6 Chat

ARSE also offers a group chat. Each project has its own group chat, where members of that specific project can discuss about issues relevant to that space. The chat is displayed in the bottom-right corner and it is shown in each view of the project (e.g. Sprint Board, Backlog, ...). The chat window is collapsible, it can be in two different states: shown (Figure 3.8) and minimized or hidden (Figure 3.9).

When the chat window is minimized, each time a new message arrives, there is a red indicator displaying the number of new unseen messages. When displayed, it shows the 10 most recent messages, offering also the possibility to search through older messages by clicking the “Load more” button which appears on top of these messages. By clicking the button, 10 more messages are loaded. The chat is also mobile-friendly, where in the same corner a floating action button appears. Through this button the chat window can be collapsed and minimized. The messages exchanged per project are persistent.

4 Done Product Backlog Items

In the following we list all the items from the original product backlog that were developed in this project. Certain stories have been moved to make the actions available to a different class of users, so that the product is more compatible with the idea of Scrum. The roles in our project are users, registered users, team members (of a project), and product owners (of a project). This list also contains stories that were modified or added to the original backlog.

All mandatory features of the original product backlog have been implemented. The bonus features (email confirmation, burndown charts) were not addressed, however, we focused on some features for a good user experience that were not required. This includes using drag and drop to change the priority of stories and a very responsive application with complete mobile compatibility.

- As a user, I can
  - register to the system with user name, email and password. (Unique usernames)
    - Users can be assigned to different projects with different roles.
  - access the web application with different kinds of devices (mobile, tablet, pc . . .).
- As a registered user, I can
  - edit my profile (name, password).
  - see the list of projects I am a member of.
  - create new projects (name and description), and become automatically the product owner of these projects.
- As a team member of a project, I can
  - access the Product Backlog.
  - access the Sprint Board.
  - access past sprints which includes their name, time interval, and story points burnt.
  - copy backlog items to the Sprint Backlog.
  - assign product backlog items to a team member.
  - add new product backlog items to the product backlog of a project.
    - product backlog items can have different statuses (new, in progress, done).
    - product backlog items can be of different types, e.g. feature, enhancement, fix.
  - see more details of a particular product backlog item.
  - change the order (priority) of product backlog items in the product backlog.
    - reorder the product backlog items using drag and drop.
    - change the scope of a running Sprint by dragging items into or out of the Sprint scope.
  - change product backlog items including their description as well as their status (New, in progress, done).
  - create new tasks in the Sprint Board (a task always belongs to a user story).
  - change tasks including their description as well as their status (New, in progress, done).
  - chat with my team members.
  - get a notification if a new chat message is received.
  - access and search chat logs.
- As a product owner of a project, I can
  - create a new Sprint (a new Sprint cannot be created if there's one already running) with name, description, time-box and initial Sprint Backlog.
  - assign registered users to a project.
  - assign roles to project members (team member, product owner).
  - change the list of statuses assignable to product backlog items in the project settings.
  - change the list of types assignable to product backlog items in the project settings.
  - close a Sprint (all done stories will be removed).
  - cancel a Sprint.

Bibliography

[1] *Node.js*. URL: `https://nodejs.org/en/` (visited on 03/06/2016).

[2] *Express.js*. URL: `http://expressjs.com/` (visited on 03/06/2016).

[3] *MongoDB*. URL: `https://www.mongodb.com/` (visited on 03/06/2016).

[4] *Angular.js*. URL: `https://angular.io/` (visited on 03/06/2016).

[5] *AngularJS Full-stack Generator*. URL: `https://github.com/angular-fullstack/generator-angular-fullstack` (visited on 03/06/2016).

[6] *Socket.io*. URL: `http://socket.io/` (visited on 03/06/2016).

[7] *Twitter Bootstrap*. URL: `http://getbootstrap.com/2.3.2/` (visited on 03/06/2016).