TECHNISCHE
UNIVERSITÄT
DARMSTADT

TK Internet Praktikum

Team Novemver - Scrum-iT (MEAN Stack Application)

Name : Shashank Singh
Matriculation Number : 2678605

Name : Mohd Danish
Matriculation Number : 2756420

Name : Ratika Sharma
Matriculation Number : 2686448

Name :Kunal Sapru
Matriculation Number : 2649867

Name : Ritika Rawat
Matriculation Number : 2651318

1. Introduction

Scrum-iT is a web implementation of the Scrum process using MEAN(MongoDb, Express, Angular and Node.js) stack.

2. Technical Documentation

2.1. Technical Architecture

2.1.1. What is MEAN?

M.E.A.N stands for

1. mongoDB
   a. Which is a nosql database and more scalable than relational databases.One of the best features of mongoDB is that it accepts and returns everything in a common format which is JSON for every transaction made in the database.
2. Express framework
   a. Express is a web application server framework for node.js
3. Angular JS
   a. Which is used to assemble all our client side code like html views, javascript, css in MVC pattern and provides a rich set of features to make cutting edge client side applications.
4. Node.js server
   a. This server fully utilizes the capabilities of javascript and is used to build fast, scalable web applications.

In our project we have used the following technologies on server and client side.

2.1.2. Server Side

S.No	Dependency	Description
1	Mongoose	mongoDB object modelling for node.js
2	Promise	To handle asynchronous calls
3	body-parser	To parse html
4	io.sockets	To implement chat feature in the application
5	Fs	To read files from file system
6	Express	Web application framework for node.js
7	Bluebird/ mongoose-q/ q-bluebird	To handle asynchronous calls

8	method-override	To override HTTP verbs

2.1.3. Client Side

S.No	Dependency	Description
1	Angular	To apply mvc pattern on client side
2	angular-route	For client side page routing
3	angular-socket-io	To implement chat module
4	angular-ui-router	For client side routing
5	angular-xeditable	To perform all CRUD (create, read, update, delete) operations in a more user friendly way in one page.
6	bootstrap	To apply responsive css
7	font-awesome	Use icons in various pages
8	highcharts	To implement sprint burndown charts feature
9	jquery	Client side scripting
10	socket.io	To implement chat application
11	socket.io-client	For chat application

2.1.4. Some important files used in the project

Server.js
This file is the entry point of the application.The file contains code for running the server.

Why we use package.json?
All server related dependencies like body-parser, mongoose, socket.io etc which we are using in the application are contained in this file.
We need to use the command “npm install” to add these dependencies in our project

Why we use bower.json?
All client side dependencies like angular-ui-router, angular-xeditable etc are contained in this file. We need to use the command “bower install” to add these dependencies in our project.

What is .gitignore file used for?
.gitignore file contains the list of files/Folders which we don’t want to commit during git push.

serverRoutes.js file

serverRoutes.js contain all files used to handle corresponding http requests.

mongoDB configuration

MongoDB configuration is defined in the file database/config/mongoDBSettings.js file. Here we define the URL and the the project database which is ‘scrumit’.

2.1.5. Database Models

Following database models have been used in our application

S.No	Schema Name	Description
1	priority	Contains priority items like high, medium, low
2	project	Contains details of a project like name, description etc.
3	role	Contains role items like Developer, Product Owner, Super User etc.
4	sprintBacklog	Contains details of a sprint backlog item like name, corresponding product backlog item, Sprint ID etc.
5	sprint	Contains information related to a sprint like name, start date, end date, corresponding project ID etc.
6	status	Contains status items like New, In progress, completed, closed etc.
7	task	Contains tasks details created by developer and information of assignee, corresponding sprint backlog item and sprint ID etc.
8	userProjectRole	This schema maps a user to the corresponding project and role.
9	user	Contains user information like first name, email etc.
10	userStory	Contains user story information like name, corresponding project ID, user story type etc.
11	userStoryType	Contains user story types like feature, enhancement, fix etc.

2.1.6. Server Requests Folder

Server requests folder (/serverRequests) contains all http GET and POST requests corresponding to various schemas.

This file contains various database models whose functions we want to access. For example, initialSetUpRequest.js file uses following database model dependencies: projectModel , roleModel, userModel, userProjectRoleModel, statusModel, priorityModel and userStoryType model.

It then uses all these models to access various functions offered by the corresponding schema like addUser(), addProject(), getProject, addUserStoryType etc.

Similarly all other serverRequest files uses the dependencies that they need.

SNo.	File Name	Description
1	initialSetUpRequest.js	Contains dependency of various database models to access their functions like addUser(), addProject() etc.
2	priorityRequests.js	Contains all the functions that are related to priority schema like addPriority(), getAllPriorities() etc.
3	projectRequests.js	Contains all functions related to project schema. For eg. getAllProjectsExceptSuperUserProject(), editProject() etc.
4	roleRequests.js	Contains all functions related to role schema like getRoleByName, addRole() etc.
5	sprintRequests.js	Contains all functions related to sprint schema like checkSprintAlreadyRunning(), getCurrentSprint() etc.
6	statusRequests.js	Contains all functions related to status schema like getStatusDocByName() etc.
7	taskRequests.js	Contains all task schema related functions.
8	userProjectRoleRequests.js	Contains all functions related to userProjectRole schema like getSelectedUserNameLIst(), getAllRolesExceptUserId(), getAllAssignedProjects(), removeUserFromProject(), modifyUserRole() etc.
9	userRequests.js	Contains all functions related to user schema like registerUser(), checkUserName(), login() etc.
10	userStoryRequests.js	Contains all functions related to userStory schema like addUserStory(), getAllUserStoriesExceptCompletedAndClosed() etc.
11	userStoryTypeRequests.js	Contains all functions related to userStoryType schema like getAllUserStoryTypes(), deleteUserStoryType() etc.

2.1.7. Important Angular Concepts

Angular controllers

Angular controllers are used to handle views to which they are attached. We use the $scope inbuilt variable to send data to the corresponding view or html file.
Apart from $scope, angular controllers can make use of other important variables like $stateParams(to get request parameters), $filter and any of the services that are exposed in the application.For example, the BacklogItemsController makes use of $scope, $filter and the

services priorityService,userStoryTypeService,statusService and backlogService variables to access the functions offered by them.

Angular Services

Angular services are used to send HTTP GET and POST requests to the server side which are handled by files in serverRequests folder explained earlier. For example, PriorityServiceModule in which priorityService is exposed, makes use of priorityRequests file in serverRequests folder to access functions related to priority schema.

Angular views

These are simple HTML files which makes use of the corresponding controller defined in the appRoutes.js file.For example.createNewSprint.html view makes use of the CreateNewSprintController which is defined in appRoutes.js

```
state('homePage.sprintSettings.createNewSprint', {
     url: "/createNewSprint",
    templateUrl: 'views/createNewSprint.html',
     controller: 'CreateNewSprintController'
   }
```

2.1.8. App.js and AppRoutes.js files

App.js file contains all angular module dependencies like controller modules, service modules and other dependencies.
Here is a list of some of the angular dependencies that we have defined in this file:

```
'ui.router','btford.socket-io','xeditable','ngRoute', 'appRoutes', 'DashboardControllerModule',
'BacklogItemsControllerModule','TaskSettingsControllerModule'
```

3. User Manual

3.1. Setting up the initial Application.

3.1.1. Describe super User Role, Product owner and different users.

Following are the steps to create the intial Application:

1. Use url “http://localhost:3009/initialSetUp”
2. Creates an initial setup of the with default users , super user and Product owner
3. Super User (email : admin@admin.com, password : admin)
4. Product Owner (email : john_cena@admin.com password : john)
5. Users (email : userN@admin.com, password : userN) where N= [0 to 9] eg. user0@admin.com has password user0
6. Super User has default project “super_user_project” and Product Owner has default project “Scrum Web App”.
7. Super user can only assign projects to Project owner.
8. Project Owner assigns roles to list of user.

** *A user needs to be a part of the project to access the dashboard.*

3.1.2. Login into Application

1. Run the Application.
2. Login Page will Appear.

3. Either login with the already created users (formed with Intial setup) or Register a new user.
4. Enter Email and password.
5. Click on “Login” button to login into the application.

3.1.3. Assign Product Owner

(Login-> Click on Home button of selected project-> click on 👤▾ icon->”Click Assign Product Owner”)

1. Super user can assign Project to Product Owner.

3.1.4. As a user

I can register to the system with user name, Email Address and password

1. Run the application, login page will appear.
2. Clickon “Register Yourself”.
3. Enter Username, First Name, Last Name, Email Address, password and Confirm Password.
4. Click on the “Register User” button.
5. Click on “Back to Login”, directed to the login page
6. Enter the registered Email Address and Password to login into the application.

3.1.5. As a team member

(Login-> Click on Home button of selected project-> click on 👤▾ icon.)

I can edit my profile (Email Address, Password)

1. Click on the “User Profile” option.
2. Enter new Username, First name, Last name and Email address to edit.
3. Click on “Update” button to save the information.

Logout from the application

(Login-> Click on Home button of selected project-> click on 👤▾ icon-> Click Logout)

1. Click on Logout .
2. The user will be redirected to the Login Page.

View Burndown Chart

(Login-> Click on Home button of selected project-> click on Burndown Chart)

1. After the Burndown Chart option is clicked.
2. A burndown chart will be displayed showing Ideal and current scenario of the Sprint.

Download Burndown chart

(*Login-> Click on Home button of selected project-> click on Burndown Chart)*

1. The burndown charts can be downloaded as pdf, jpeg, svg and pdf.

Create Backlog Items

(Login-> Click on Home button of selected project-> click on Product Backlog Settings -> Create New Backlog Items→ Add New User Story)

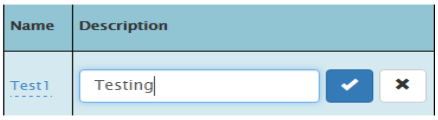
1. Click on Add New User Story button to add user story.

2. Enter the User Story Name and User Story Description.
3. Click on “choose an option”,a dropdown will appear.
4. Select priority
5. Click ✓ to save the priority.
6. Click on choose an option to select the “User Story Type”.
7. Click ✓.
8. Click on “Save User Story” button to save .
9. Click on “Close” button to close the dialog box.
10. Backlog item is added to the list.

Edit and Update Backlog Items

(Login-> Click on Home button of selected project-> click on Product Backlog Settings -> Create New Backlog Items-> Select the backlog item)

1. User can edit Name, Description,User Story Type and Priority by clicking .
2. A box will appear
3. Enter the information.
4. Click ✓ to save the information.



Name	Description
Test1	Testing ✓ ✖

Delete Backlog Items

(Login-> Click on Home button of selected project-> click on Product Backlog Settings -> Create New Backlog Items-> Select Backlog Item)

1. Select the user story.
2. Click on the Delete .
3. The selected user story is deleted.

View Completed Backlog Items

(Login-> Click on Home button of selected project-> click on Product Backlog Settings->Completed Backlog Item)

1. User can view Name, Description,Created By and Status of completed backlog item.

View Close Backlog Item

(Login-> Click on Home button of selected project-> click on Product Backlog Settings->Completed Backlog Item)

1. Select the Backlog item .
2. Click on Close.
3. Then click on “View Closed Backlog Items”.
4. All the completed and cancelled backlog items will be viewed.

Chat Module

(Login-> Click on Home button of selected project->Chat)

1. Enter the text.
2. Click “Send” button.
3. The text message will be broadcasted to every registered user.

3.1.6. As a developer of a project

Task Settings uses the angular-xeditable feature and developer can create, edit, delete and view all tasks in a single page.

Also, we can select sprint backlog items from the drop-down menu, in this way multiple tasks can be created against a single backlog item.

By having all these functionalities in a single page it is similar to a typical task board.

- We can also assign tasks to ourselves or to different developers using this feature.
- Status of the task and effort can also be added using this feature.
- Once all the tasks corresponding to a single backlog items are set to status "completed", the status of the sprint backlog item and the corresponding product backlog item changes to "completed", it will then appear in the completed section of the product backlog module. The product owner can then close the product backlog item.
- Once all sprint backlog items are completed and then closed by the product owner, the sprint status is set to "closed" and it can then be viewed in the closed section of the "sprint settings" module.

3.1.7. As a Product owner of a project

Create new projects

(Login-> Click on Home button of selected project->Master Table-> Project)

1. Click on "Add Project" button.
2. Enter Project Name and Project Description.
3. Click on "Save Project" button.
4. Click on the "Close" button to close the dialog box .

Edit / Update New Project

(Login-> Click on Home button of selected project->Master Table-> Project-> Click project item to edit)

1. Click on the corresponding "Project Name".
2. Edit it and click on ✓
3. Click on the corresponding "Project Description".
4. Edit it and click on ✓.

Delete New Project

(Login-> Click on Home button of selected project->Master Table-> Project->Select Project)

1. Click Delete button to delete the corresponding Project Name and Project Description.

Assign registered users to a project assign roles to project members

(Login-> Click on Home button of selected project->Click 👤▾ -> Project Settings)

1. Click on "choose and assign a role".
2. Select role from drop down list.
3. Click on ✓ to assign role to user.

View Assigned Members

(Login-> Click on Home button of selected project->Click 👤▾ -> Project Settings)

1. Click on "Goto Assigned Project Members.
2. Assigned User List will be displayed.

Edit/Update Assigned Members

1. Click on the assigned role.
2. Select the role to assign from drop down list.
3. Click on ✓ button to save.

Delete User from Project

(Login-> Click on Home button of selected project->Click 👤▾ -> Project Settings-> click “Goto Assigned Project Members”)

1. Click the “Delete User from Project” button to delete the corresponding assigned user.
2. The corresponding assigned user will be deleted.

Create a sprint

(Login-> Click on Home button of selected project->Sprint Settings-> click “Create New Sprint”)

Note: **User cannot create a new sprint if a sprint is already running*

1. Enter Sprint details Sprint Name, Description ,select starting &end date and Estimated Effort.
2. Click on the “Create New Sprint”.

View Current Sprint

(Login-> Click on Home button of selected project->Sprint Settings-> click “View Current Sprint”)

1. Current Sprint information will be displayed.

Sprint History

(Login-> Click on Home button of selected project->Sprint Settings-> click “Sprint History”)

1. Previous sprint information will be displayed.

Master Tables

Theses tables are used to add new values to Priority,Project, Role,Status and userStory type

Create New Priority

(Login-> Click on Home button of selected project->Master Table-> click “Priority”->”Add Priority”)

1. Enter Priority Name.
2. Enter Priority Description.
3. Click on “Save Priority” button to save.
4. Click on “Close” button to close the pop up window

Edit/Update New Priority

(Login-> Click on Home button of selected project->Master Table-> click “Priority”)

1. Select the priority for Priority Name and Priority Description by clicking over it.
2. Click on ✓ to save the priority.

Delete Priority

(Login-> Click on Home button of selected project->Master Table-> click “Priority”)

1. Click on “Delete” button to delete the corresponding Priority Name and Priority Description.

Create New Role

(Login-> Click on Home button of selected project->Master Table-> click "Role"->"Add Role")

1. Click on "Add Role button".
2. Enter Role Name.
3. Enter Role Description.
4. Click on "Save Role" button to save the information.
5. Click on "Close" button to close the pop up window.

Edit/Update Role

(Login-> Click on Home button of selected project->Master Table-> click "Role")

1. Click on the Role Name to edit.
2. Click on ☑ to save it.
3. Click on the Role Description to edit.
4. Click on ☑ to save it.

Delete Role

(Login-> Click on Home button of selected project->Master Table-> click "Role")

1. Click on the "Delete" button to delete the corresponding Role Name and Role Description.

Create New Status

(Login-> Click on Home button of selected project->Master Table-> click "Status"->"Add Status")

1. Click on "Add Status" button.
2. Enter Status Name.
3. Enter Status Description.
4. Click on "Save Status" button to save the information.
5. Click on "close "button to close the pop up window.

Edit/Update Status

(Login-> Click on Home button of selected project->Master Table-> click "Status")

1. Click on the status under the Status Name that is required to be edited.
2. Click on ☑ to save it.
3. Click on the Description under the Status Description that is required to be edited.

Delete Status

(Login-> Click on Home button of selected project->Master Table-> click "Status")

1. Click on the "Delete" button to delete the corresponding Status Name and Status Description.

Create New User Story Type

(Login-> Click on Home button of selected project->Master Table-> click "User Story type"->"Add User Story type")

1. Click on "Add User Story type" button.
2. Enter User story Type.
3. Enter User Story Description.
4. Click on "Save User Story" button to save the information.
5. Click on the "Close" button to close the pop up window.

Edit/Update User Story Type

(Login-> Click on Home button of selected project->Master Table-> click "User Story type")

1. Click on value under "UserStoryType Name" and "UserStoryType Description".
2. Change the value to Fix,Enhancement or Feature.
3. Click on ✓ to save it.

Delete User Story Type

(Login-> Click on Home button of selected project->Master Table-> click "User Story type")

1. Click on the "Delete" button to delete the corresponding UserStoryType Name and UserStoryType Description.