ID2207 HT23 Modern Methods in Software Engineering (50928)

Final Project - Group 8

Dominika Drela, Eugen Lucchiari Hartz

GitHub Repo with code and Readme file: https://github.com/Eugenius0/SEP

User Stories

(when we say employee in these user stories we mean the employees that are authorised)

- Login: Any employee in the SEP company can access the system through the login screen where she/he enters her/his user name and password. After verification user is logged in

Time Estimate: GUI 0.5 hour, Logic 1 hour

- Authorization mechanism, split from Login: based on the logged in user's authorization level, she/he will be able to access different functionalities, adapted for all the four different workflows.

Time Estimate: GUI 5 hours Logic 6 hours

- View Event Request List: Employee should be able to see a list of created event requests

Time Estimate: GUI 2 hours Logic 4

- View Event Request Workflow: Employee should get an overview of the different steps of the workflow and the point at which each event request currently is

Time Estimate: GUI 3 hours Logic 1 hour

- Navbar: Employee should be able to navigate through the four different main workflows

Time Estimate: GUI 1 hour Logic 1 hour

- Create event request form: Employee should be able to create an event request by filling out a form and submit it.

Time Estimate: GUI 3 hours Logic 3 hours

- Register client form: Employee should be able to register a client by filling out a form and submit it.

Time Estimate: GUI 1 hour Logic 2 hours

- Event request details form for SCS: SCS should be able to specify details about an event request which is filled during the first business meeting with the client, ususally by the SCS officer.

Time Estimate: GUI 2 hours Logic 2 hours

- Task distribution overview: Employee should be able to get an overview of the assigned and commented tasks

Time Estimate: GUI 2 hours Logic 1 hour

- Create task: Employee should be able to create a task by filling out a form and submit it.

Time Estimate: GUI 1 hour Logic 3 hours

- Recruitment request overview: Employee should be able to get an overview of the recruitment requests and their status

Time Estimate: GUI 2 hours Logic 3 hours

- Create recruitment request: Employee should be able to create a recruitment request by filling out a form and submit it.

Time Estimate: GUI 1 hour Logic 3 hours

- Financial request overview: Employee should be able to get an overview of the financial requests and their status

Time Estimate: GUI 2 hours Logic 3 hours

- Create financial request: Employee should be able to create a financial request by filling out a form and submit it.

Time Estimate: GUI 1 hour Logic 3 hours

- Error page: In case a wrong URL is used the employee gets redirect to an error page

Time Estimate: GUI 0.5 hour Logic 0.5 hour

- Select items and get corresponding functionalities: Employee should be able to select an item of a list and get accordingly access to different functionalities which are then related to the selected element. Needed for all the workflows.

Time Estimate: GUI 4 hours Logic 6 hours

(All of these user stories have been implemented, more or less complete, there is always room for improvement)

Release Planning

User Story Name	Value	Risk
Login	High	Medium
Authorization mechanism	High	High
View Event Request List	Medium	Low
View Event Request Workflow	Medium	Medium
Navbar	High	Low
Create event request	High	Low
Register client form	High	Low
Event request details form for SCS	High	Medium
Task distribution overview	High	Medium
Create task	High	Medium
Recruitment request overview	High	Medium
Create recruitment	High	Medium
Financial request	High	Medium
Create financial request	High	Medium
Error page	Low	Low
Select items and get corresponding functionalities	High	High

For choosing user stories for the releases the idea is of first focusing on the combinations of (high, low), (high, medium), (high, high), (medium, low), (medium, medium), …

First release would include accordingly the following user stories (60% of all, so 10):
Navbar
Create event request form
Register client form
Login
Event request details form for SCS
Task distribution overview
Create task
Recruitment request overview
Create recruitment request
Financial request overview

Rest of user stories for release 2

Iteration Planning

First Iteration:
Navbar
Create event request form
Register client form
Login

Second Iteration:
Event request details form for SCS
Task distribution overview
Create task

Third Iteration:
Recruitment request overview
Create recruitment request
Financial request overview

Metaphor

Metaphor	System
SEP	The event management company
CS	Customer Service
SCS	Senior Customer Service
FM	Financial Manager
AM	Administration Manager
PM	Production Manager
SM	Service Manager
ST	Sub-Team
SHR	Senior Human Ressources
Rejected	A request that has been declined
Pending	A request that is waiting for further processing (rejection, approvement etc.)
Approved	A request that has been accepted
SCS approved	A request that has been accepted by the Senior Customer Service
Feedback	A request that has been reviewed and received feedback
AM approved	A request that has been accepted by the Administration Manager

Scheduled Meeting	Request for which a client meeting is scheduled
Photography, Music, Graph Design, Decorations, Network Support	Different sub-teams that are responsible for a specific field
Commented	A request that has been reviewed and received a comment
Event request	A request for managing an event
Task distribution	Tasks that get assigned to a sub-team and get comment by the assigned team
Staff recruitment/recruitment request	A request to ask for additional resources, employees
Financial Requests	A request to ask for additional financial resources

Description of your test-driven pair programming process and applied refactoring. Also describe how well you managed to estimate what should be done in each iteration.

We had a specific concept for our pair programming sessions.
Before actually starting we did a stand up meeting where we both explained what we did the day before and what we plan to do today and if we had problems with anything specific. In case of identified problems we first pair programmed on that, otherwise we discussed what are the next most important steps that need to be implemented and pair programmed on that. After defining what to work on next we started the actual session. On person is the driver which is actively writing code and the other person is the navigator which reviews the code as it's being written and focus on the bigger picture, code quality, and suggest improvements. Every 30 minutes we switched the roles. We repeated this until reaching the goals that we defined at the beginning of the session.
Our team's estimation accuracy improved over time as we gained a better understanding of the complexity of our tasks. In the initial iterations, our estimates were less accurate, but as we continued to work together, our ability to estimate what should be done in each iteration improved. Accordingly, the iterations became more predictable and successful.

Acceptance Test

Successful login

Test Case Name	Login
Expected actions	1. Navigate to login page 2. Enter email: user5@sep.com 3. Enter password: password5 4. Click on submit

Expected results	The system should redirect the user to the Event Requests page. On the top right it should be displayed logged in as Logged in as: PM. So given credentials are valid
Test result	Successful

Authorization mechanism

Test Case Name	Authorization mechanism
Expected actions	1. Logged in as PM 2. Click on Staff Recruitment at the top in the nabber to navigate to the staff recruitment page 3. Log out and log in again, but this time as for example Customer Service 4. Navigate again to the Staff recruitment page
Expected results	The system should display a Create Recruitment Request Button to the Production Manager. A user that has for example Customer Service as a role should not be able to see this button
Test result	Successful

Stand-up meeting report

Meeting Date	28.10.2023
Participant	Eugen Lucchiari Hartz, Dominika Drela
Meeting Notes	1. Summary of our activities in the previous day: - authentication and authorization mechanism - Navbar - Event request list 2. Today expected actions: - Create event request page - Tasklist - Create Task page - Staff recruitment page - Create recruitment request - Financial requests page - Create financial request

Comments	Focus on efficiency and problem solving, ask for help if stuck

Meeting Date	29.10.2023
Participant	Eugen Lucchiari Hartz, Dominika Drela
Meeting Notes	1. Summary of our activities in the previous day: - Create event request page - Tasklist - Create Task page - Staff recruitment page - Create recruitment request - Financial requests page - Create financial request 2. Today expected actions: - Selected items feature - Improve some parts of the authorization mechanism - Do user acceptance tests to test proper functionality of the system
Comments	Focus on efficiency and problem solving, ask for help if stuck

Comparison between this approach and the object oriented analysis and design approach - your feedback

Agile is more adaptable, customer-centric, and suitable for projects with changing or uncertain requirements, but it may not fit well for all types of projects. Object-Oriented Analysis and Design, on the other hand, provides a more structured approach with detailed planning and documentation, making it more appropriate for projects with well-defined requirements. The choice between the two approaches depends on the nature and specific needs of the project.

Appendix

Notes out of made commits
- create login UI
- create login authenticate mechanism
- create EventRequestList
- create eventRequest Workflow
- create navbar
- authorization mechanism
- create event request form

- register client form
- event request details form for scs
- task distribution
- user authorization for task distribution
- create task page
- error page
- Staff recruitment page
- create recruitment request
- financial request page
- create financial request
- selected Item for all workflows

needed improvements:
- change detection of selected item so that it checks the id of the selected item and not title
- selected item should only sometimes appear at the top (does not make sense to reject a already rejected task for example)
- fix style for selected items
- add database and API so that queries and functionalities can actually be executed, to replace mockData
- adapt task distribution workflow for service manager and corresponding sub-teams
- make it fully responsive