User stories

1. **Login story**

Login

“As an employee of the company, I want to be able to authenticate myself using the username and password which was uniquely provided to me, so that I could perform operations that my role authorizes me to do.”

Time Estimate:
GUI: 0.5 hours
Logic: 1 hour

2. **Event request story**

Event request

“As a client I want to contact the available customer service officer to be able to request an event, so that my event request could be documented and reviewed by the management of the company, allowing the event to take place if it is approved.”

Time Estimate:
GUI: 1.75 hours
Logic: 2.75 hours

- Initiate event request story
  Split from: “Event Request story”

Initiate event request

“As a customer service officer I want to record clients requests in a new form so that they could be reviewed in detail by the management.”

Time Estimate:
GUI: 0.5 hours
Logic: 1 hours

- SCS review event request story
  Split from: “Event Request story”

SCS review event request

“As the senior customer service officer I want to be able to review the incoming event requests from the customer service officers so that I could approve or deny them before forwarding them for further review.”

Time Estimate:
GUI: 0.25 hours

Logic: 0.25 hours

- Financial feedback of event request story
  Split from: “Event Request story”

Financial feedback of event request

“As the financial manager I want to receive the financial information related to the specific event request, so that I could add feedback on the budget, which would inform the department manager of my concerns with the feasibility of the event.”

Time Estimate:
GUI: 0.25 hours
Logic: 0.50 hours

- Final event request reviewal story
  Split from: “Event Request story”

Final event request reviewal

“As the administrative manager I want to view the full event request and feedback associated with it so that I could make an informed decision on whether or not our company should proceed with the event.”

Time Estimate:
GUI: 0.25 hours
Logic: 0.50 hours

- Informing client story
  Split from: “Event Request story”

Informing client

“As the senior customer service officer I want to inform the customer of the company's decision regarding event organization, so that the customer would be updated with the latest information and if the event is approved, be able to provide us with more information, such as preferences and other details of interest.”

Time Estimate:
GUI: 0.5 hours
Logic: 0.5 hours

3. **Tasks distribution story**

Tasks distribution

“As the department manager I want to be able to assign my staff to different tasks within my department, so that they could work on them and there would be no confusion of responsibilities.”

Time Estimate:
GUI: 1.5 hours
Logic: 4.25 hours

- Create task story
  Split from: “Tasks distribution story”

Create task

“As a department manager I want to be able to create new tasks for the event, by entering relevant details of customers' requests in the form, so that these tasks could easily be understood and assigned to members of my staff.”

Time Estimate:
GUI: 0.5 hours
Logic: 2 hours

- Assigns task story
  Split from: “Tasks distribution story”

Assigns task

“As a department manager I want to assign tasks to a specific employee, by picking them from the list of available workers, so that he could carry out the execution of this request.”

Time Estimate:
GUI: 0.25 hours
Logic: 1 hours

- Plan activities story
  Split from: “Tasks distribution story”

Plan activities

“As a staff member I want to make plans based on the task i was assigned to and add comments for the department manager to read, where necessary, so that the current issues could be remedied and the possibility of future issues limited.”

Time Estimate:
GUI: 0.5 hours
Logic: 1 hours

- Change status story
  Split from: “Tasks distribution story”

Change status story

“As a department manager I want to change the status of the event to be “open”, so that it is noticeable that this event is open for planning.”
Time Estimate: GUI: 0.25 hours Logic: 0.25 hours

4. **Staff recruitment story**

Staff recruitment “As a department manager i want to request recruitment of more human resources, so that the planned event could be executed within the intended time frame.”
Time Estimate: GUI: 2.25 hours Logic: 2.75 hours

- Review staff availability story
  Split from: “Staff recruitment story”

Review staff availability “As the department manager I want to observe the availability of my staff, so that I would know if there is a need to hire more workers, to perform the tasks necessary for the event to be successful.”
Time Estimate: GUI: 1 hour Logic: 1 hours

- Initiate recruitment request story
  Split from: “Staff recruitment story”

Initiate recruitment request “As a department manager I want to fill in a recruitment request form, so that the human resources team would be notified that there is insufficient amount of workers to accomplish the tasks.”
Time Estimate: GUI: 0.5 hours Logic: 1 hour

- Recruitment story
  Split from: “Staff recruitment story”

Recruitment

“As a human resources team member I want to be notified if there is any need for more workforce, so that I could hire/outsource work based on the information provided by the manager of the department within the request form.”

Time Estimate:
GUI: 0.5 hours
Logic: 0.5 hours

- Change status story
  Split from: “Staff recruitment story”

Change status

“As a department manager I want to change the status of the event to be “in progress” if there are no recruitment or financial problems associated with the event, so that it is noticeable that this event is in progress.”

Time Estimate:
GUI: 0.25 hours
Logic: 0.25 hours

5. **Financial request story**

Financial request

“As the department manager I want to register a financial request, if there are issues with the current budget of the event, so that the financial manager could negotiate the budget issues with the customer.”

Time Estimate:
GUI: 0.75 hours
Logic: 1.5 hours

- Request budget negotiation story
  Split from: “Financial request story”

Request budget negotiation

“As the department manager I want to create a financial request, providing detailed description of the issue at hand, so that the financial manager could deal with budget problems.”

Time Estimate:
GUI: 0.5 hours
Logic: 1 hours

- Negotiates budget issues story

Split from: “Financial request story”

Negotiates budget issues “As the financial manager I want to receive a budget negotiation request, detailing the shortcomings of the current financial state, so that the customer could be informed and issue solved, before continuing work on event planning.”
Time Estimate: GUI: 0.25 hours Logic: 0.5 hours

- Change status story

  Split from: “Financial request story”

Change status “As a department manager I want to change the status of the event to be “in progress” if there are no recruitment or financial problems associated with the event, so that it is noticeable that this event is in progress.”
Time Estimate: GUI: already discussed in Staff recruitment story Logic: already discussed in Staff recruitment story

Release planning

User Story	Value	Risk	Time	Release
Login Story	High	Low	1.5	1
Initiate event request story	High	Medium	1.5	1
SCS review event request story	Medium	Low	0.5	2
Financial feedback of event request story	Medium	Medium	1.25	2
Final event request reviewal story	High	Medium	1.25	1
Informing client story	Low	Medium	1	2
Create task story	High	Medium	2.5	1
Assigns task story	High	High	1.25	1
Plan activities story	Medium	High	1.5	2
Change status story	Low	Low	0.5	2
Review staff availability story	Medium	High	2	2
Initiate recruitment request story	High	Medium	1.5	1
Recruitment story	High	Low	1	1
Change status story	Low	Low	0.5	2
Request budget negotiation story	High	Medium	1.5	1
Negotiates budget issues story	High	Low	0.75	1
Change status story	Low	Low	0	2
			20	

	High Value	Medium Value	Low Value
High Risk	1	2	0
Medium Risk	5	1	1
Low Risk	3	1	3

Release	Combination (value, risk)	Sum Time
1	(high, high), (high, medium), (high,low)	12.75
2	(medium, high), (medium, medium), (medium, low), (low, medium), (low, low)	7.25

Iteration planning

Release 1

User Story	Value	Risk	Time	Iteration	Assigned To	Time Taken
Login Story	High	Low	1.5	1	Spoorthi	1.75
Initiate event request story	High	Medium	1.5	1	Ralfs	2
Final event request reviewal story	High	Medium	1.25	1	Spoorthi	1
Create task story	High	Medium	2.5	2	Spoorthi	2.5
Assigns task story	High	High	1.25	2	Spoorthi	1.5
Initiate recruitment request story	High	Medium	1.5	3	Spoorthi	1
Recruitment story	High	Low	1	3	Spoorthi	1.25
Request budget negotiation story	High	Medium	1.5	3	Ralfs	1.5
Negotiates budget issues story	High	Low	0.75	3	Ralfs	0.5
			12.75			13

	High Value	Medium Value	Low Value
High Risk	1	0	0
Medium Risk	5	0	0
Low Risk	3	0	0

Iteration	Sum Time
1	4.25
2	3.75
3	4.75

Release 2

User Story	Value	Risk	Time	Iteration	Assigned To	Time Taken
SCS review event request story	Medium	Low	0.5	1	Spoorthi	1
Financial feedback of event request story	Medium	Medium	1.25	1	Ralfs	1
Informing client story	Low	Medium	1	2	Ralfs	1
Plan activities story	Medium	High	1.5	2	Ralfs	1.5
Change status story	Low	Low	0.5	1	Ralfs	0.25
Review staff availability story	Medium	High	2	3	Ralfs	1.5
Change status story	Low	Low	0.5	3	Spoorthi	0
Change status story	Low	Low	0	3	Ralfs	0
			7.25			6.25

	High Value	Medium Value	Low Value
High Risk	0	2	0
Medium Risk	0	1	1
Low Risk	0	1	3

Iteration	Sum Time
1	2.25
2	2.5
3	2.5

Metaphors:

User- Employee using the system
Client- Person for whom an event could be organized
Staff- Usersunder the same department
Role- Defines users access rights and responsibilities
Event Request- Event requested by the client, but yet not approved
Event- Approved event request
Task- Organizational activity assigned to a staff members
Comment- Discussion about the task and issues related to it (done by the user)
Financial Request- Request to negotiate budget issues with the client
Recruitment Request- Request to hire more workforce
Seed- Default data used during the software development for testing

The system is a pizza delivery- When the client wants a pizza, he/she could request it by calling the pizzeria, then the pizzeria checks if it can perform the request (do they have the resources/manpower), later the delivery route is planned, finally if no issues are encountered the pizza is delivered.

Working test driven

When working in a test driven approach (following TDD) we started by thinking of what the system should do and before implementing any functionality we created the tests that would assure us that the method, which we are to implement, would work as intended. In the beginning the tests would not compile, but when working on the tested method we could see that tests start to compile, but they would still fail. Based on this we add functionality solving a single test case at a time, possibly refactoring the solution along the way, if it could be simplified or changed completely, resulting in a better product. Finally, after all of the tests are running (including the previously created tests for other methods), the work on the function is finished, since its goals, which are described in the tests, have been achieved.

Pair programming

Whenever possible we worked following the pair programming aspect of the extreme programming. When doing so we both worked on the same code, with one of us taking the role of “driver” and the other one being the “navigator”. The roles would be changed for each use case, so that there would not be for example just a single person writing the code, this is described in the “release planning document”, with the task being assigned to the “driver”, while the “navigator” is not mentioned, since we were just a 2 member team.

Refactoring

Throughout the development the code was refactored multiple times, mostly to reduce the complexity of the solution, improve the readability or to make it more efficient.

Some of the most notable changes, that took place after the implementation of the code are:

1. Changed solution from using permissions to roles, as permissioned solution would be harder to use and roles did just as good of a job for the task at hand.
2. Fixing the bug with the solution of not being able to close the application until the user has successfully logged in.
3. Improved the seed function, to allow for better testing
4. Making the user interface more responsive and informative for the user

Estimates

Estimates for each iteration were done surprisingly well, besides the first iteration, where we didn't account for the need to establish the architecture and so forth, leading to us underestimating the work it would take to accomplish the tasks in time. This, however, was just a slight miscalculation in an otherwise smooth operation, allowing us to finish all of our goals for this project and delivering a usable product.

Acceptance tests

Test Case Name	Start New Recruitment Test
Assumptions	Login user is an HR Manager who has permission to start new recruitment request
Expected Actions	1. Open Application 2. Login with relevant credentials 3. In the main screen, click on Recruitment 4. New window opens with all the current Recruitment requests and a button to create a new Recruitment 5. Click on Create Recruitment Request button 6. New window opens with a recruitment request form 7. Select Contract Type 8. Select requesting department 9. Input Years of experience required 10. Input job title 11. Optionally include job description 12. Click on the Create button
Expected Results	The manage recruitment window should show all current recruitment requests with the new updated one.
Test Result	Success

Test Case Name	Start New Event Request Test
Assumptions	Login user is a Customer Service Officer who has permissions to create a new event request
Expected Actions	1. Open Application 2. Login with relevant credentials 3. In the main screen, click on event requests 4. A new window opens 5. Click on Create Event Request button 6. Input record number 7. Select the client 8. Input Event type 9. Select the from and to dates of the event 10. Input the number of attendees 11. Input the budget for the event 12. Click on Save
Expected Results	A pop up confirms that a new event request was created
Test Result	Success

Daily meetings

Meeting Date	12 October 2020
Participants	Spoorthi, Ralfs
Meeting Notes	1. Summary of Yesterday's Activities a. Login Form added b. Request Event added. c. Add financial feedback. 2. Today's expected actions a. Starting SCS approve story b. Inform client story c. Form to view events 3. Problems a. Some work beyond story requirements which were not expected
Written by	Spoorthi

Meeting Date	13 October 2020
Participants	Spoorthi, Ralfs
Meeting Notes	1. Summary of Yesterday's Activities a. SCSO approve story b. Administration manager review of event request c. Inform client of decision 2. Today's expected actions a. Create task b. Plan activities c. Change status 3. Problems a. Changed permission based system to role based
Written by	Ralfs

Meeting Date	14 October 2020
Participants	Spoorthi, Ralfs
Meeting Notes	1. Summary of Yesterday's Activities a. Create task story b. Assigns task story c. Plan activities story d. Change status story 2. Today's expected actions a. Review staff availability story
Written by	Spoorthi

Comparison of the current approach with object oriented analysis and design approach:

Using the current approach, the time taken during the design phase of the project was significantly less than the Object oriented analysis, as we were able to come up with the development plan quickly and start implementing without any unnecessary overhead. However, in this approach, the relationship between the objects were not completely clear as we headed into the implementation phase. But, the encountered problems could be easily handled due to the flexibility of this approach. Considering the simplistic nature of the problem, this solution does not require a heavy methodology like object oriented analysis as it would be overkill, it being better suited for projects requiring considerable amount of documentation.