ID2207
Modern Methods in Software Engineering

Homework 3 - Group 19

September 2021
Royal Institute of Technology

Aksel Uhr
Abdullah Abdullah

1. Identification of objects

Entity objects of first use case “initiate event request”

Employee entity
Storing data of employees.

Request entity
Storing data of requests.

Budget entity
Storing data of budgets.

Boundary objects of the first use case/scenario

Login form (view)
Including login form and buttons where the user can log in.

Request view
Including a request form, create and send buttons where a customer service employee can initiate requests.

Budget view
Where an employee of the financial department can fill a form with budget conclusions and send it further to the administration manager.

Notification / replies
Acknowledgements sent from the system to a certain employee. In the first two diagrams, we use replies as notifications (dashed arrows going to left).

Control objects of the first use case/scenario

Login controller
Connects the login form with the system and lets the user login. In this case, we assume that authentication and authorization is handled by the login controller. A more detailed version of authorization is described below in the document.

Request controller
Forwards the data which is filled in by the customer service, and creates a new request in the request entity. Also responsible for fetching correct budget data out of the request and sending it further.

Budget controller
Forwards the data which is filled in by the financial manager.

Entity objects of second use case “complete event request”

Employee entity

Storing data of employees.

Request entity

Storing data of requests.

Budget entity

Storing data of budgets.

Schedule entity

For employees storing data of schedules.

Event specification entity

Storing data of event specifications.

Boundary objects of the second use case/scenario

Login form (view)

Including login form and buttons where the user can log in.

Request view

Including a request form, create and send buttons where a customer service employee can initiate requests.

Budget view

Where an employee of the financial department can fill a form with budget conclusions and send it further to the administration manager.

Notification / replies

Acknowledgements sent from the system to a certain employee. In the first two diagrams, we use replies as notifications (dashed arrows going to left).

Schedule view

Views scheduling of the employees and date filtering options.

Event specification view

Including a form and buttons where the employee can fill with event details and requirements.

Control objects of the second use case/scenario

Login controller

Connects the login form with the system and lets the user login. In this case, we assume that authentication and authorization is handled by the login controller. A more detailed version of authorization is described below in the document.

Request controller
Forwards the data which is filled in by the customer service, and creates a new request in the request entity. Also responsible for fetching correct budget data out of the request and sending it further.

Budget controller
Forwards the data which is filled in by the financial manager.

Schedule controller
Responsible for communication between the scheduling entity and its view. Forwards data to and from the user and the scheduling entity

Event specification controller
Forwards new event information, creates and inserts new specifications and notifies the user.

Entity objects of the third use case "requestAdditionalRecruitment"

Authentication and Authorization information entity.
For the system SEP we have different employees in the system and each is associated with specific duties and role in the system. So to track each employee role and login credentials an entity object is required to persist the authentication and authorization details of each employee.

Department Employee information entity.
Employee information entity tracks each employee data like current projects the employee is involved in, tasks assigned to each employee and availability of employees (schedule). This will be accessed by the production manager to view the team availability for a required project.

Event Specification Information entity (Event Requirements)
Event specification information entity has all the data of different requirements of the projects/events from different SEP Departments for all the projects/events that have been registered in the system.

Company Resources Information entity (Need of hiring new employee or outsourcing the task)
Company resources entity has the information about lack of resources in the organization that needs to be filled in to complete certain project requirements.

Company Budget Information entity (To hire resource or outsource project)
Company budget entity has the information about the current budget allocation of the company for resources (able to hire or outsource a task).

Boundary objects of the third use case/scenario

Login form
The system has to display a login form to authenticate and authorize different users of the system.

Event Requirements View
All the requirements for different events registered in the system can be displayed through this view.

Department Employee Information View (availability)
Managers can have a view of their team members' activity and their schedule. From this view they can decide whether they require a new resource for the project or not.

Request Resource form
The system will display request additional resource application forms to production and service department managers.

Notification View
The system will send notification when the process for resolvement of resources is started.

Control objects of the third use case/scenario

Login controller
Login controller is the glue between Login form and Authentication controller.

Authorization controller
Authorization controller connects login form and authorization entity logic.

Event Data Controller
To get the event details from the entity, the event data controller will handle the requests for the boundary object.

Employee Data Controller
To get details about different employees from the employee information entity, the employee data controller will handle requests.

Resource Controller
To know about company open positions that need to be filled, users will have to access it through the resource controller from the company resource entity.

Budget Controller
To know about the company current budget and see whether it allows for new hiring or outsourcing a project budget controller will fetch data from the company budget information entity.

Request Resource Controller
Request resource controller forwards the data received from login request boundary object to update company resource entity about the resource hiring status to be in progress.

Notification Controller
Notification controller is responsible for sending notifications within the system when a new resource hiring process is started.

Entity objects of fourth use case “createSubTeamTasks”

Authentication and Authorization information entity.
For the system SEP we have different employees in the system and each is associated with specific duties and role in the system. So to track each employee role and login credentials an entity object is required to persist the authentication and authorization details of each employee.

Department Member information entity.
Employee information entity tracks each employee data like current projects the employee is involved in, tasks assigned to each employee and availability of employees (schedule). This will be accessed by the managers to view the team availability for a required project.

Event Specification Information entity (Event Requirements)
Event specification information entity has all the data of different requirements of the projects/events from different SEP Departments for all the projects/events that have been registered in the system.

Department Requirements Entity
To store the requirements of events from specific departments.

Team Tasks Entity
Team tasks entity will have all the information about the tasks assigned to each member and also the edited info by the team members.

Boundary objects of fourth use case/scenario

Login form
The system has to display a login form to authenticate and authorize different users of the system.

Event Requirements View
All the requirements for different events registered in the system can be displayed through this view.

Department Requirements View

This view will be used by the managers to fill in requirements from clients projects related to their departments.

Department Member Information View (availability)
Managers can have a view of their team members' activity and their schedule.

Assign SubTeam Tasks form
The system will display task assignment form to managers where they can assign the different tasks to their team members.

Edit Task Form
Each team member will have functionality to edit their assigned tasks with their planned activities and additional requests for resources if needed for example extra budget required.

Notification View
The system will send notification when tasks are assigned to team members and edited tasks visible to managers.

Control objects of fourth use case/scenario

Login controller
Login controller is the glue between Login form and Authentication controller.

Authorization controller
Authorization controller connects login form and authorization entity logic.

Event Data Controller
To get the event details from the entity, the event data controller will handle the requests for the boundary object.

Department Requirements Controller
To store the data for specific department requirements for an event department requirements controller will be used to handle requests from the entity.

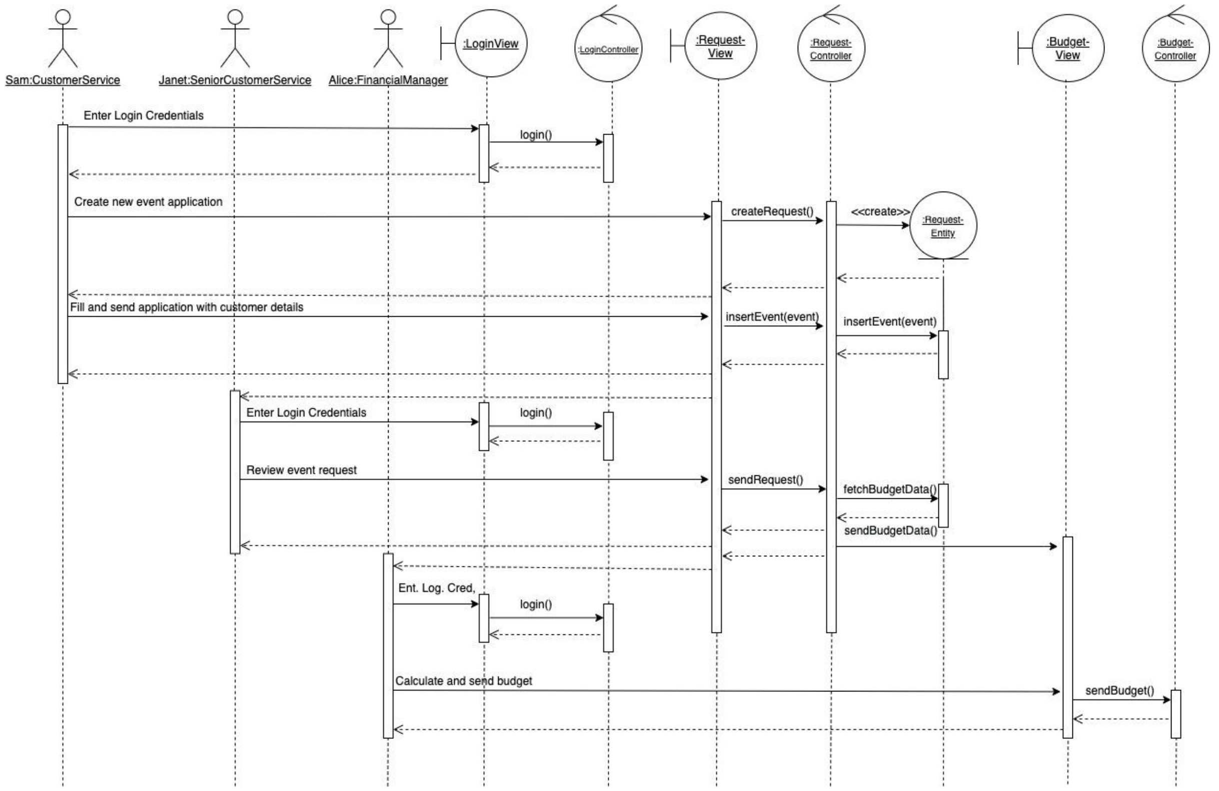
Department Member Data Controller
To get details about different employees from the employee information entity, the employee data controller will handle requests.

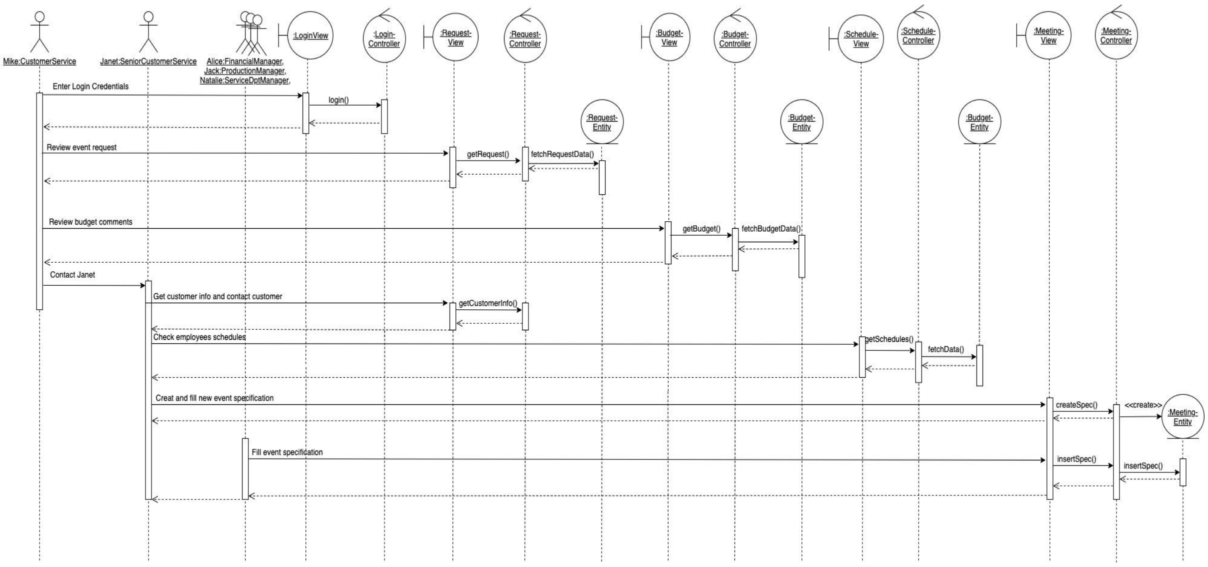
Task Assign Controller
To store the assignment of tasks for each member Task Assign controller will take info from the form and store it in Task Assign entity.

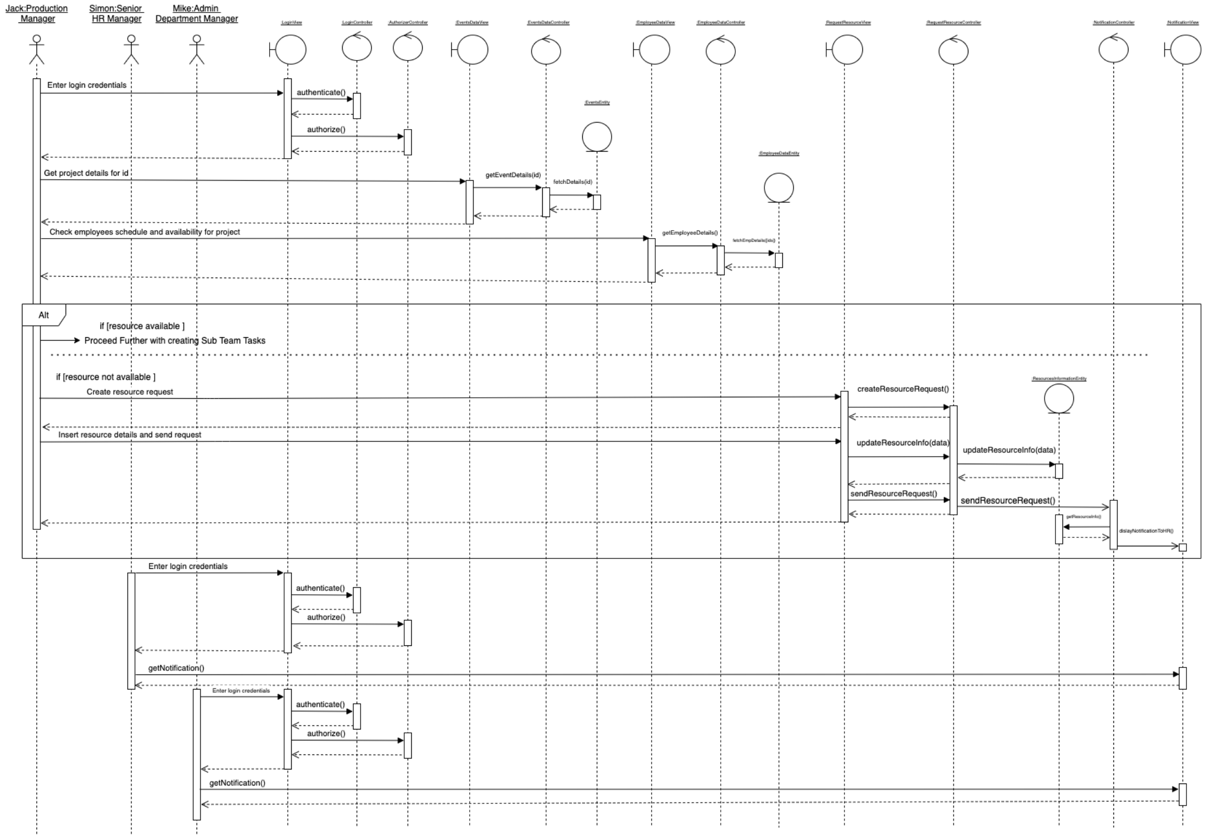
Notification Controller
Notification controller is responsible for sending notifications within the system when a new resource hiring process is started.

2. Sequence Diagrams

Based on the use cases: initiateEventRequest, completeEventRequest, requestAdditionalRecruitment and createSubTeamTasks, using the identified objects above

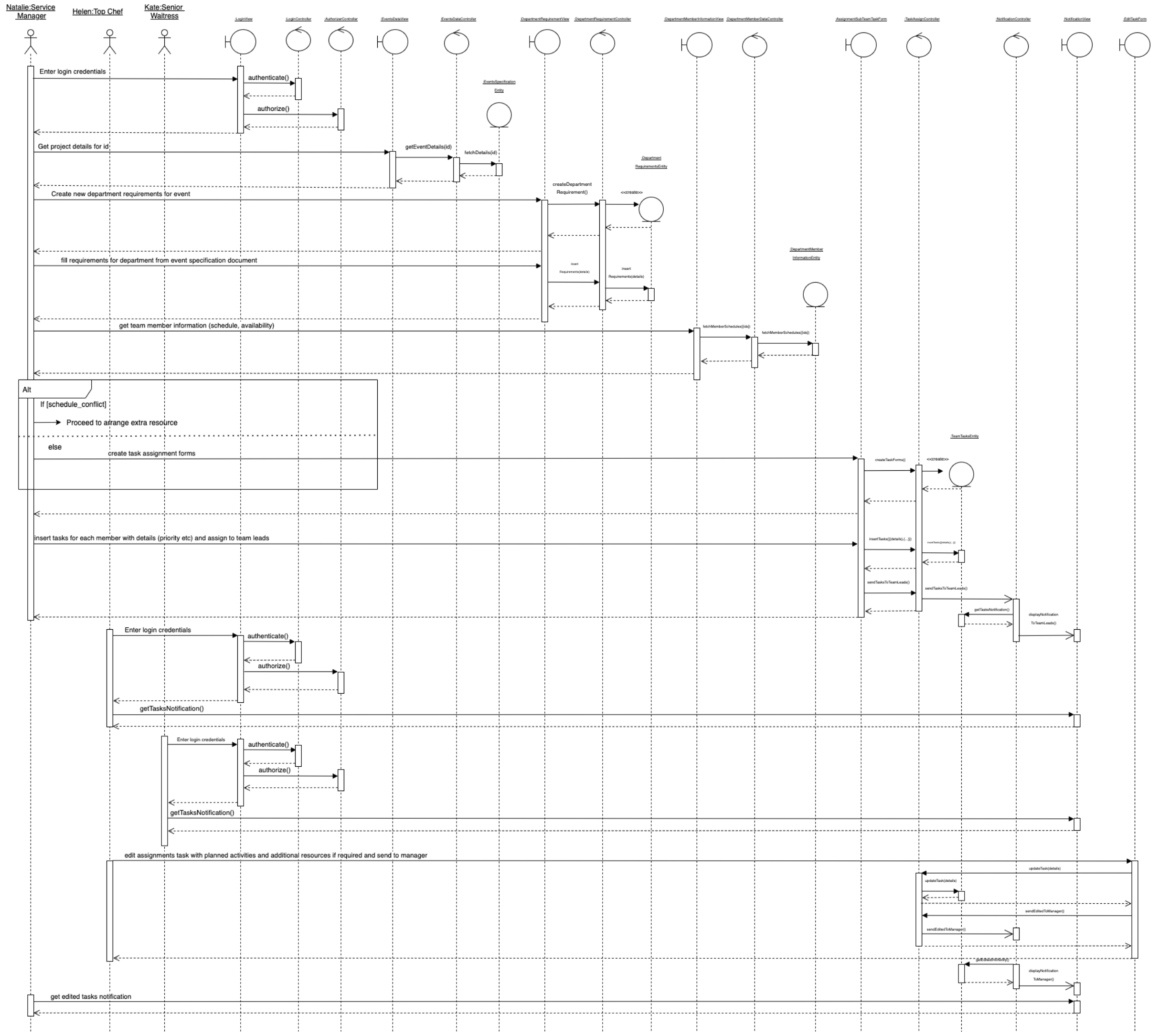






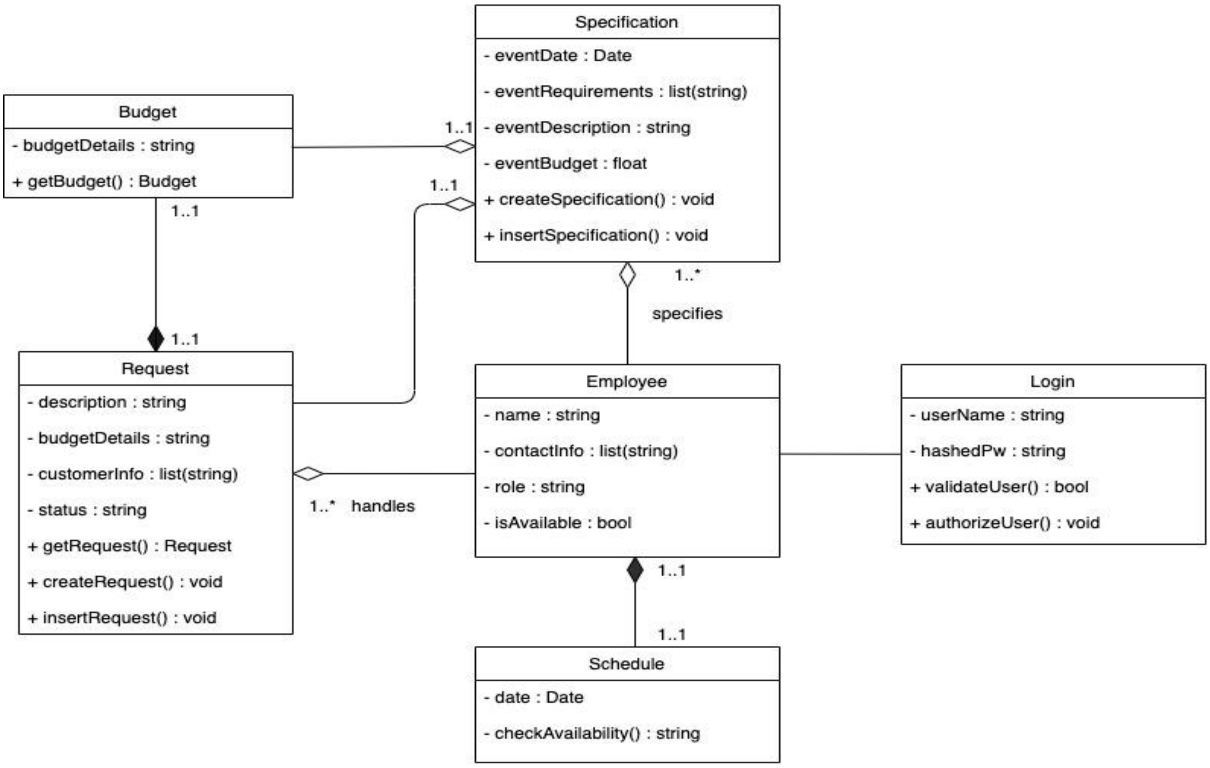
September 2021
Royal Institute of Technology

Aksel Uhr
Abdullah Abdullah

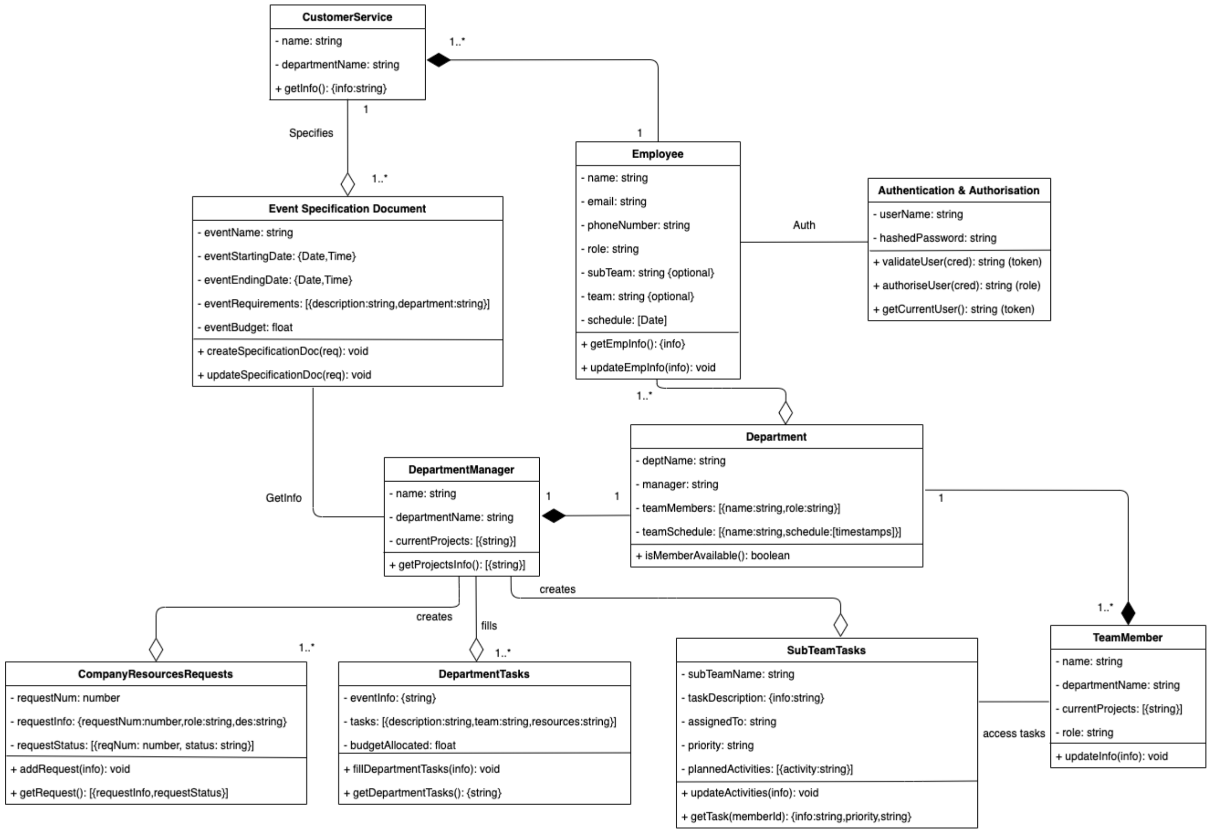

Natalie:Service Manager
Helen:Top Chef
Kate:Senior Waitress
:LoginView
:LoginController
:AuthorizerController
:EventsDataView
:EventsDataController
:DepartmentRequirementView
:DepartmentRequirementController
:DepartmentMemberInformationView
:DepartmentMemberDataController
:AssignmentSubTeamTaskForm
:TaskAssignController
:NotificationController
:NotificationView
:EditTaskForm
Enter login credentials
authenticate()
authorize()
:EventsSpecification Entity
Get project details for id
getEventDetails(id)
fetchDetails(id)
:Department RequirementsEntity
Create new department requirements for event
createDepartment Requirement()
<<create>>
fill requirements for department from event specification document
insert Requirements(details)
insert Requirements(details)
:DepartmentMember InformationEntity
get team member information (schedule, availability)
fetchMemberSchedules([ids])
fetchMemberSchedules([ids])
Alt
If [schedule_conflict]
Proceed to arrange extra resource
else
create task assignment forms
:TeamTasksEntity
createTaskForms()
<<create>>
insert tasks for each member with details (priority etc) and assign to team leads
insertTasks([{details},{...}])
insertTasks([{details},{...}])
sendTasksToTeamLeads()
sendTasksToTeamLeads()
getTasksNotification()
displayNotification ToTeamLeads()
Enter login credentials
authenticate()
authorize()
getTasksNotification()
Enter login credentials
authenticate()
authorize()
getTasksNotification()
edit assignments task with planned activities and additional resources if required and send to manager
updateTask(details)
updateTask(details)
sendEditedToManager()
sendEditedToManager()
getEditedInfoNotify()
displayNotification ToManager()
get edited tasks notification

3. Class Diagrams

Class diagram of the request handling business process.



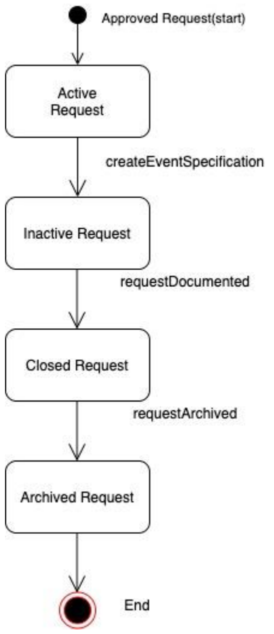
Class diagram of the recruitment and subteam tasks business processes.


CustomerService
- name: string
- departmentName: string
+ getInfo(): {info:string}
1..*
1
Specifies
1..*
1
Employee
- name: string
- email: string
- phoneNumber: string
- role: string
- subTeam: string (optional)
- team: string (optional)
- schedule: [Date]
+ getEmpInfo(): {info}
+ updateEmpInfo(info): void
Auth
Authentication & Authorisation
- userName: string
- hashedPassword: string
+ validateUser(cred): string (token)
+ authoriseUser(cred): string (role)
+ getCurrentUser(): string (token)
Event Specification Document
- eventName: string
- eventStartingDate: {Date,Time}
- eventEndingDate: {Date,Time}
- eventRequirements: [{description:string,department:string}]
- eventBudget: float
+ createSpecificationDoc(req): void
+ updateSpecificationDoc(req): void
1..*
GetInfo
DepartmentManager
- name: string
- departmentName: string
- currentProjects: [{string}]
+ getProjectsInfo(): [{string}]
1
1
Department
- deptName: string
- manager: string
- teamMembers: [{name:string,role:string}]
- teamSchedule: [{name:string,schedule:[timestamps]}]
+ isMemberAvailable(): boolean
1
creates
creates
fills
1..*
1..*
1..*
CompanyResourcesRequests
- requestNum: number
- requestInfo: {requestNum:number,role:string,des:string}
- requestStatus: [{reqNum: number, status: string}]
+ addRequest(info): void
+ getRequest(): [{requestInfo,requestStatus}]
DepartmentTasks
- eventInfo: {string}
- tasks: [{description:string,team:string,resources:string}]
- budgetAllocated: float
+ fillDepartmentTasks(info): void
+ getDepartmentTasks(): {string}
SubTeamTasks
- subTeamName: string
- taskDescription: {info:string}
- assignedTo: string
- priority: string
- plannedActivities: [{activity:string}]
+ updateActivities(info): void
+ getTask(memberId): {info:string,priority,string}
access tasks
TeamMember
- name: string
- departmentName: string
- currentProjects: [{string}]
- role: string
+ updateInfo(info): void

1

4. State Chart Diagrams

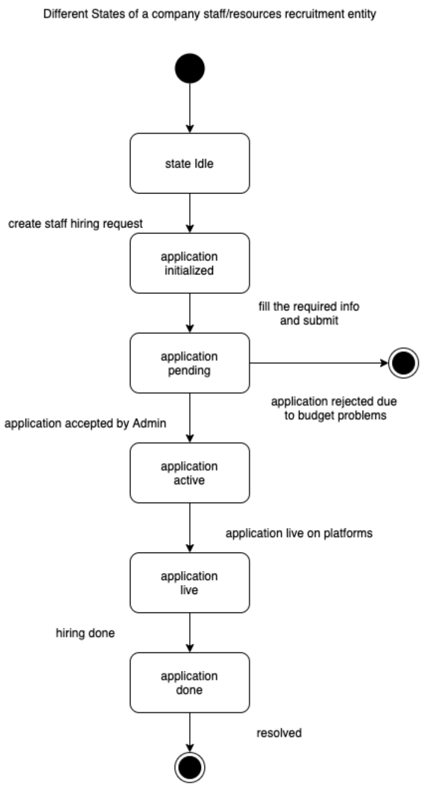
Different states of a request object.



September 2021
Royal Institute of Technology

Aksel Uhr
Abdullah Abdullah

Different States of company resource object


Different States of a company staff/resources recruitment entity
state Idle
create staff hiring request
application initialized
fill the required info and submit
application pending
application rejected due to budget problems
application accepted by Admin
application active
application live on platforms
application live
hiring done
application done
resolved