Experimental Material for the paper entitled

Cross-Domain and Cross-Language Reuse: Insights into Challenges and Opportunities

Table of Contents

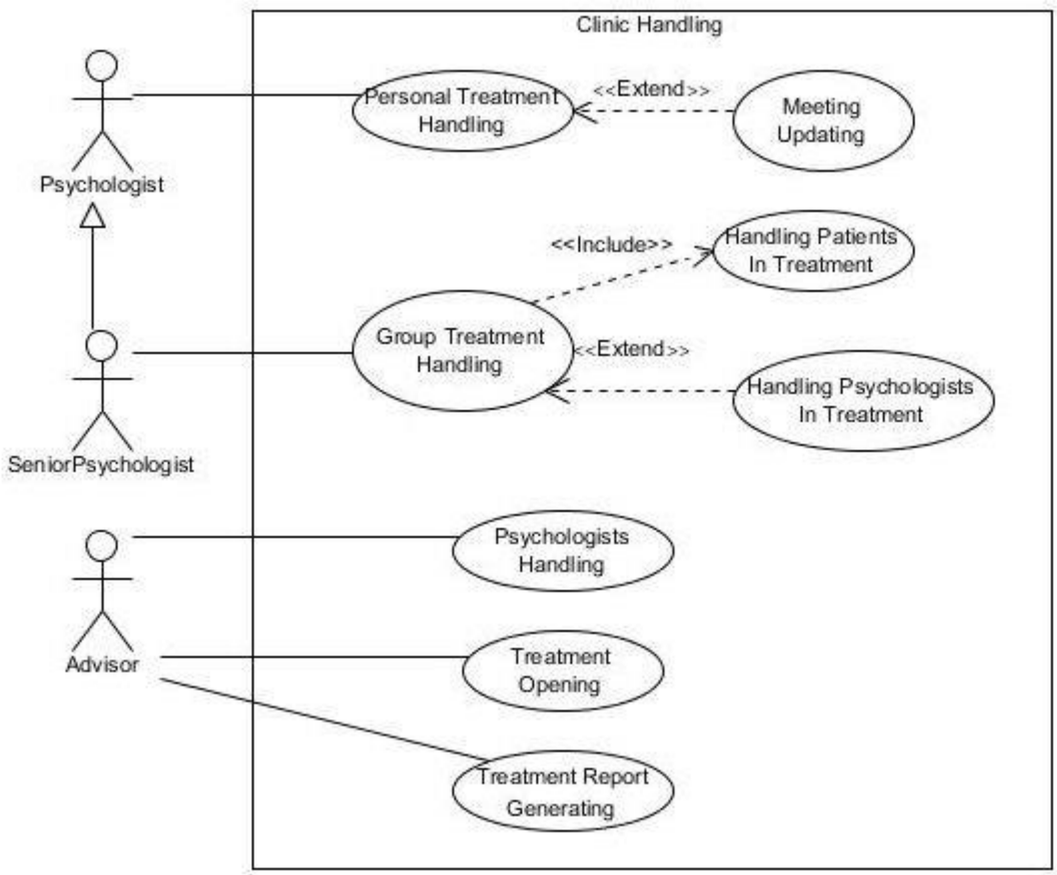
Comment: The requirements were originally given in Hebrew (important terms were given both in Hebrew and English). Here, they are translated to English. The models were originally created in English. The screenshots are provided as they were given in the experiments (in Hebrew).

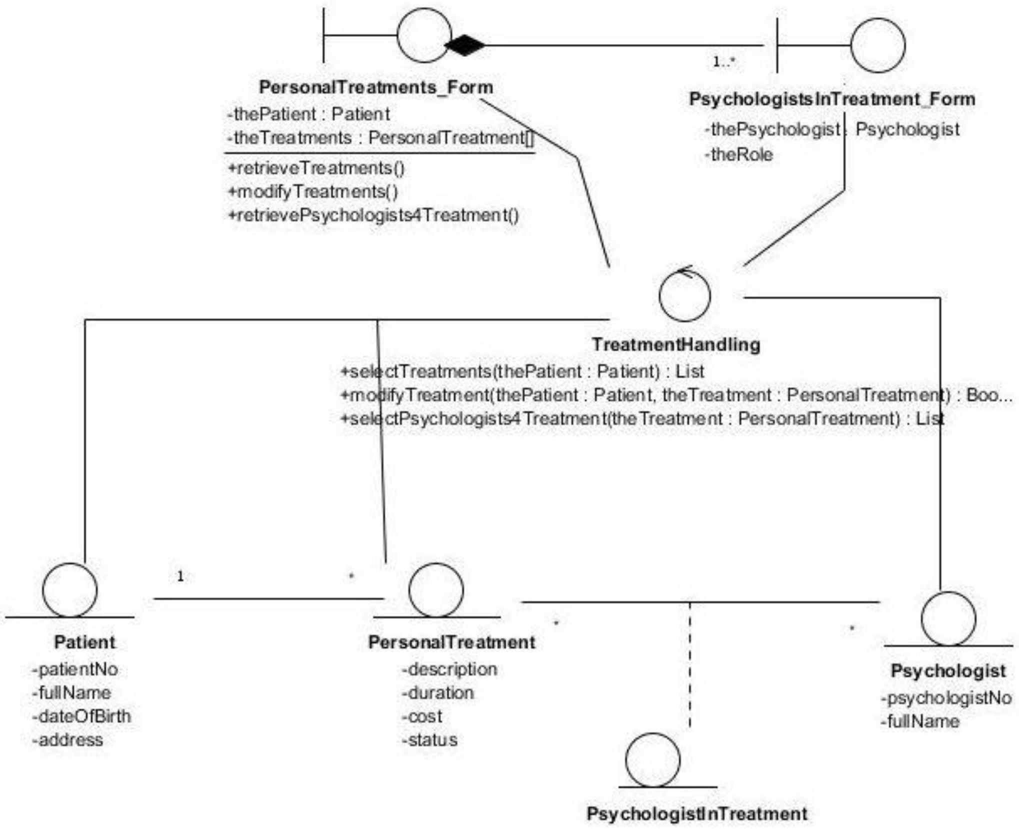
1 Experiment 1

1.1 Part 1

In this part, you will be required to model in ***UML use case and class diagrams*** a system for ***monitoring students' difficulties during their studies*** (hereinafter, *the new system*). For this purpose you can reuse parts of *an existing system* model for ***managing psychologist clinics***. Reuse can be done as-is, or with updates, additions or omissions. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements).

1.1.1 The model of the existing system




PersonalTreatments_Form
-thePatient : Patient
-theTreatments : PersonalTreatment[]
+retrieveTreatments()
+modifyTreatments()
+retrievePsychologists4Treatment()
1..*
PsychologistsInTreatment_Form
-thePsychologist : Psychologist
-theRole
TreatmentHandling
+selectTreatments(thePatient : Patient) : List
+modifyTreatment(thePatient : Patient, theTreatment : PersonalTreatment) : Boo...
+selectPsychologists4Treatment(theTreatment : PersonalTreatment) : List
1
*
Patient
-patientNo
-fullName
-dateOfBirth
-address
PersonalTreatment
-description
-duration
-cost
-status
*
*
Psychologist
-psychologistNo
-fullName
PsychologistInTreatment

1.1.2 The requirements from the new system

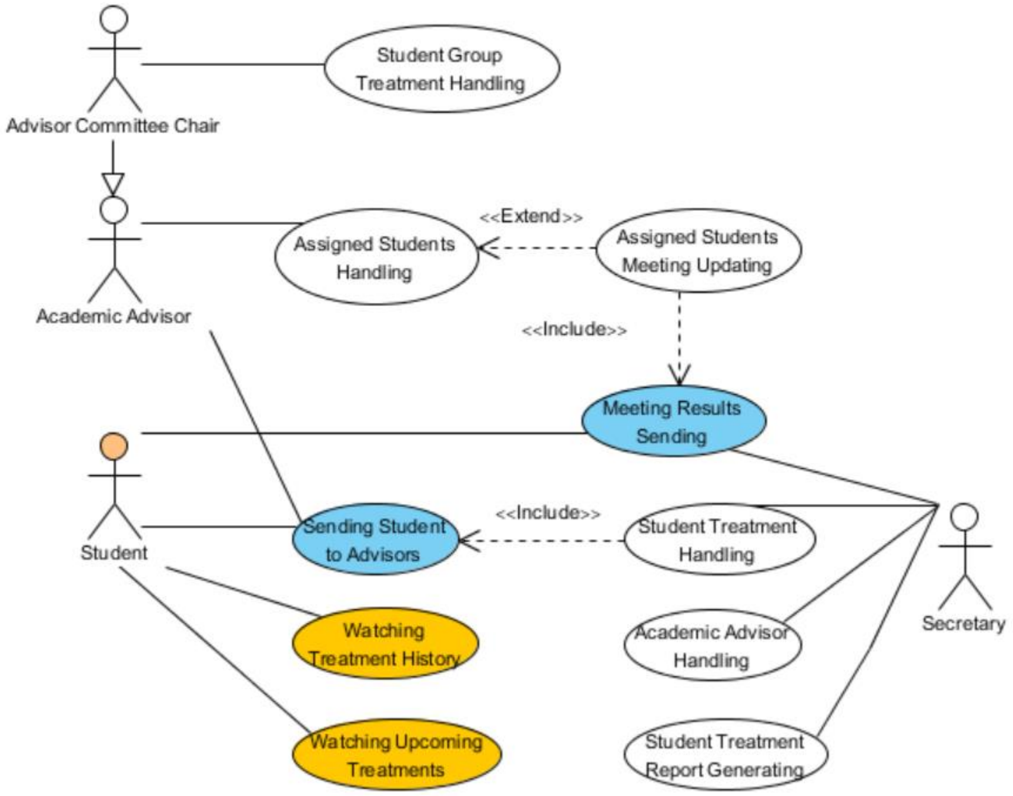
1. The users of the new system are students and academic advisors, as well as secretaries.
2. The secretariat can manage details on advisors, open treatment for students who have difficulties in their studies (a process that includes referral of students to the appropriate advisors while updating both parties) and create a periodic treatment report.
3. Advisors can manage the difficulties of their associated students, including updates on holding meetings if necessary. After updating a meeting, the results of the meeting will be sent automatically to both the student and the secretariat.
4. The chairman of the Academic Advisors Committee, who is an academic advisor, can also manage group consulting that will be relevant to students with similar difficulties.
5. Students will be able to view the history of their treatments, as well as future treatments planned.
6. A student can have several treatments, with each treatment performed by

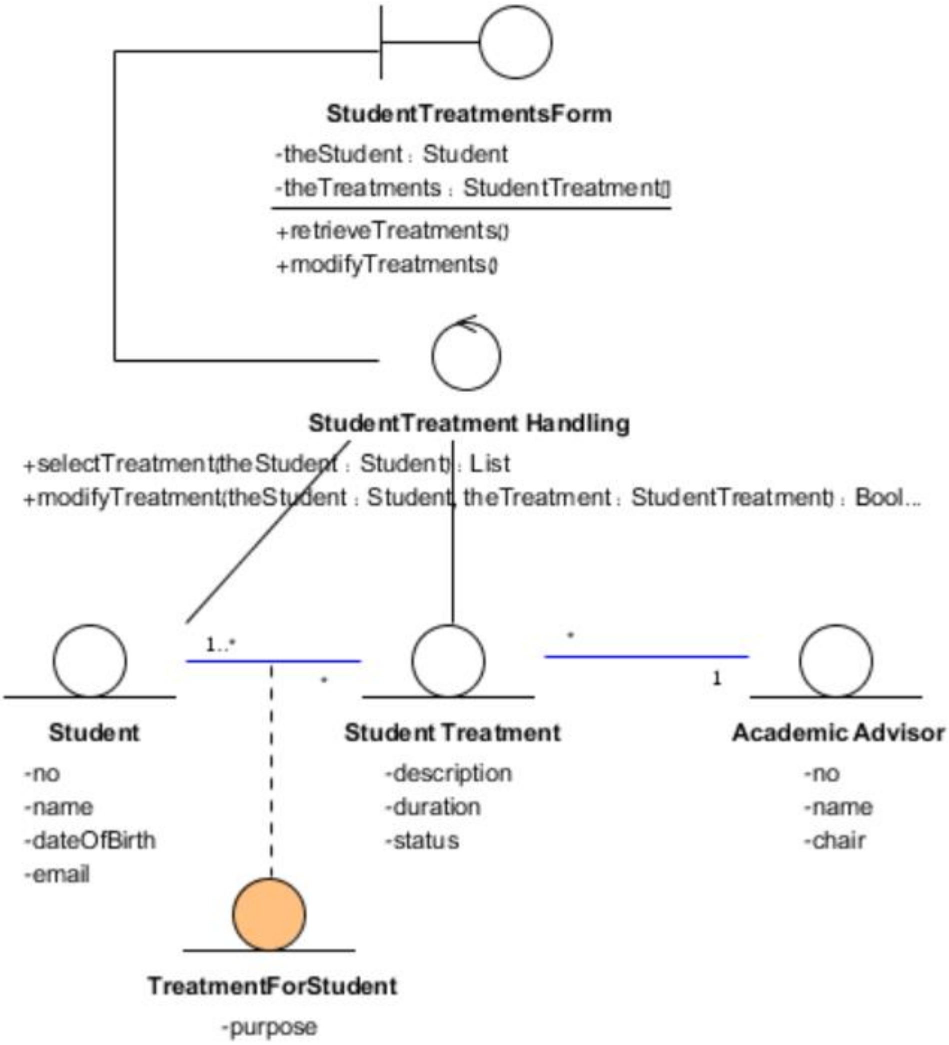
one academic advisor.

7. A treatment can be shared by a number of students with similar difficulties. For each student, the system will maintain the purpose of the treatment for him/her. The information stored for a student is her/his number, name, date of birth and email address.
8. When conducting the treatments, the academic advisor is presented with information about the student and his/her treatments. The advisor can retrieve the student's treatments (even if performed by other academic advisors) and update the treatment.

1.1.3 Possible model of the new system (expected solution)

<Addition is orange, Modification in blue, use-as-is in black>

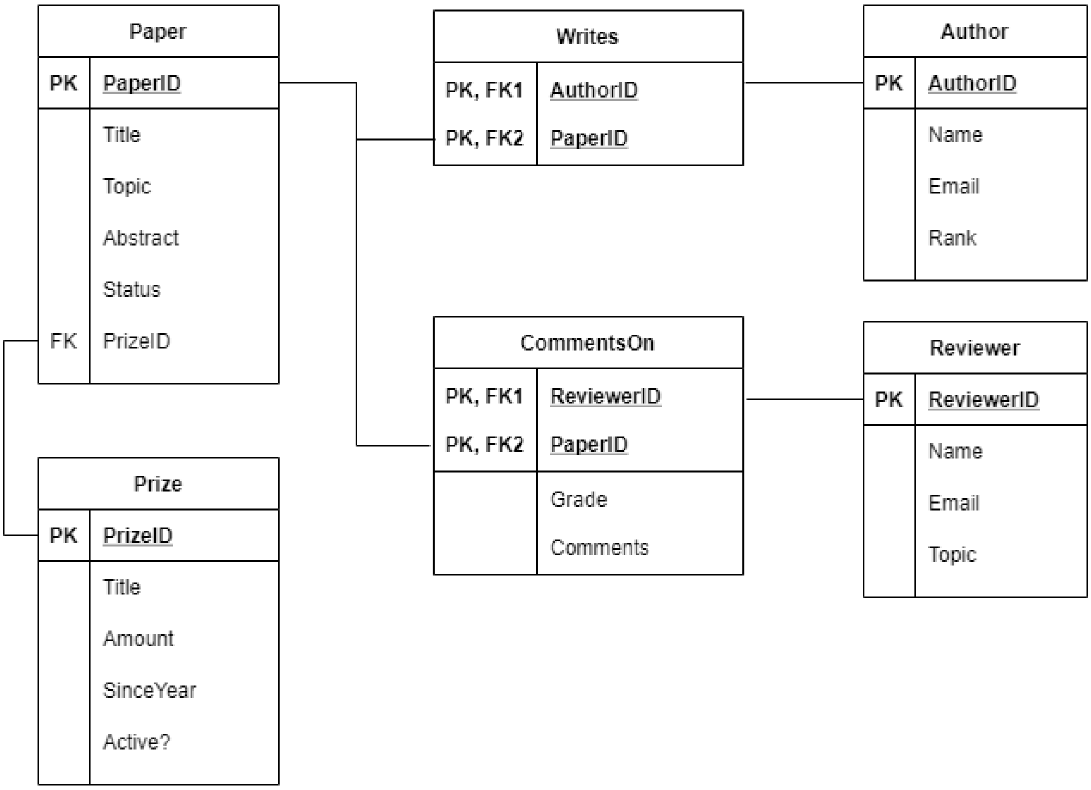

Advisor Committee Chair
Student Group Treatment Handling
Academic Advisor
Assigned Students Handling
<<Extend>>
Assigned Students Meeting Updating
<<Include>>
Meeting Results Sending
Student
Sending Student to Advisors
<<Include>>
Student Treatment Handling
Secretary
Watching Treatment History
Academic Advisor Handling
Watching Upcoming Treatments
Student Treatment Report Generating

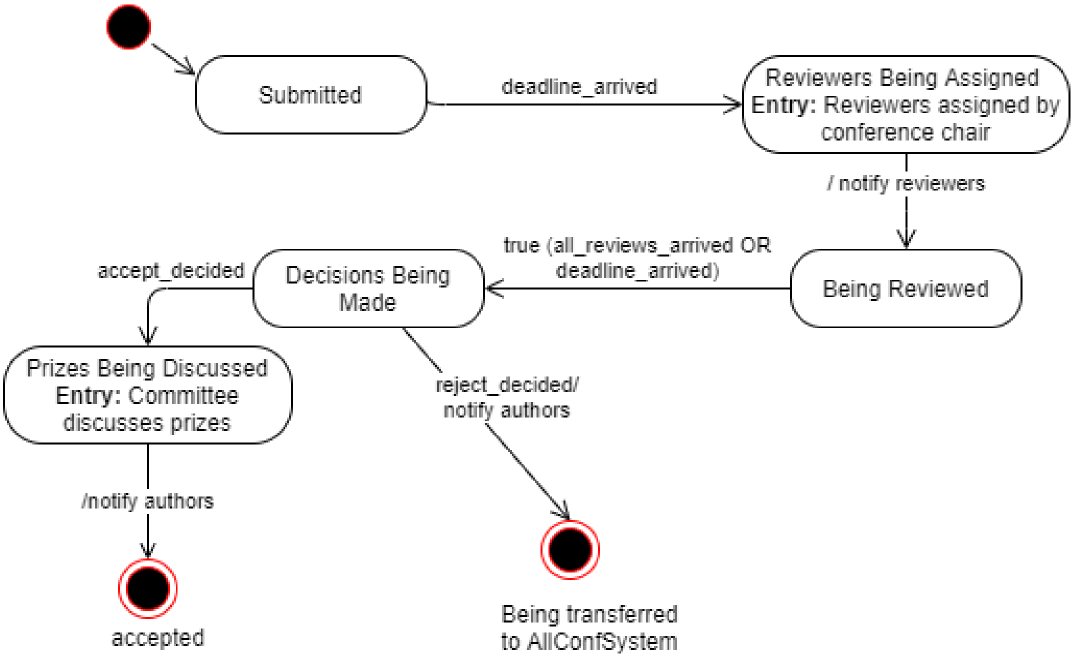

StudentTreatmentsForm
-theStudent : Student
-theTreatments : StudentTreatment[]
+retrieveTreatments()
+modifyTreatments()
StudentTreatment Handling
+selectTreatment(theStudent : Student) : List
+modifyTreatment(theStudent : Student, theTreatment : StudentTreatment) : Bool...
1..*
*
*
1
Student
-no
-name
-dateOfBirth
-email
Student Treatment
-description
-duration
-status
Academic Advisor
-no
-name
-chair
TreatmentForStudent
-purpose

1.2 Part 2

In this part, you will be required to model in ***ER and DF diagrams*** the next generation of a ***paper review management system*** (hereinafter *CovidDesign2021*, *the new system*). To do this, you can reuse parts of *an existing system* model for ***paper review of a previous conference – physicalDesign2019***. Reuse can be as-is, or with updates, additions or omissions. *Note: the existing system and the new one are modeled in different modeling languages*. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements) and that you have answered the requirements of each section (in the required language).

1.2.1 The model of the existing system




Submitted
deadline_arrived
Reviewers Being Assigned
Entry: Reviewers assigned by conference chair
/ notify reviewers
Being Reviewed
true (all_reviews_arrived OR deadline_arrived)
Decisions Being Made
accept_decided
Prizes Being Discussed
Entry: Committee discusses prizes
/notify authors
accepted
reject_decided/ notify authors
Being transferred to AllConfSystem

1.2.2 The requirements from the new system

1. Similarly to PhysicalDesign2019, CovidDesign2021 manages the processes of reviewing papers for the conference from the submission to the stage of accepting or rejecting them (while transferring the rejected papers to the AllConfSystem system).

Unlike PhysicalDesign2019, CovidDesign2021 allows:

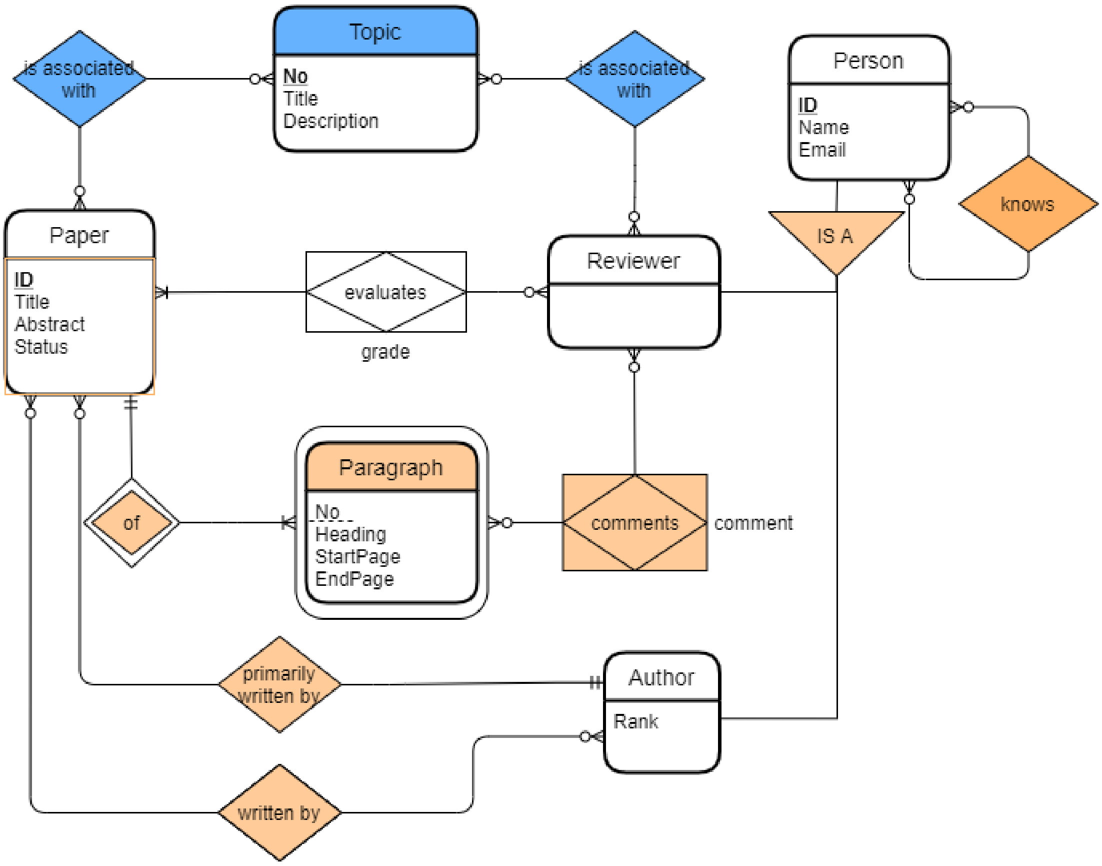
2. A paper is submitted by its authors and consists of paragraphs. The features of papers and authors are as they are in PhysicalDesign2019.
3. There is a distinction between a primary author (one) and secondary authors of a paper.
4. A paper is reviewed by reviewers. Reviewers' features are as their features in PhysicalDesign2019.
5. Submission of papers and associating reviewers are done according to a defined list of topics determined by the conference chair. A paper can be associated with several topics and so can a reviewer. Each topic has a title and a description.
6. There can be familiarity relationships between reviewers and/or authors that should be maintained. These relationships are reported by the reviewers.

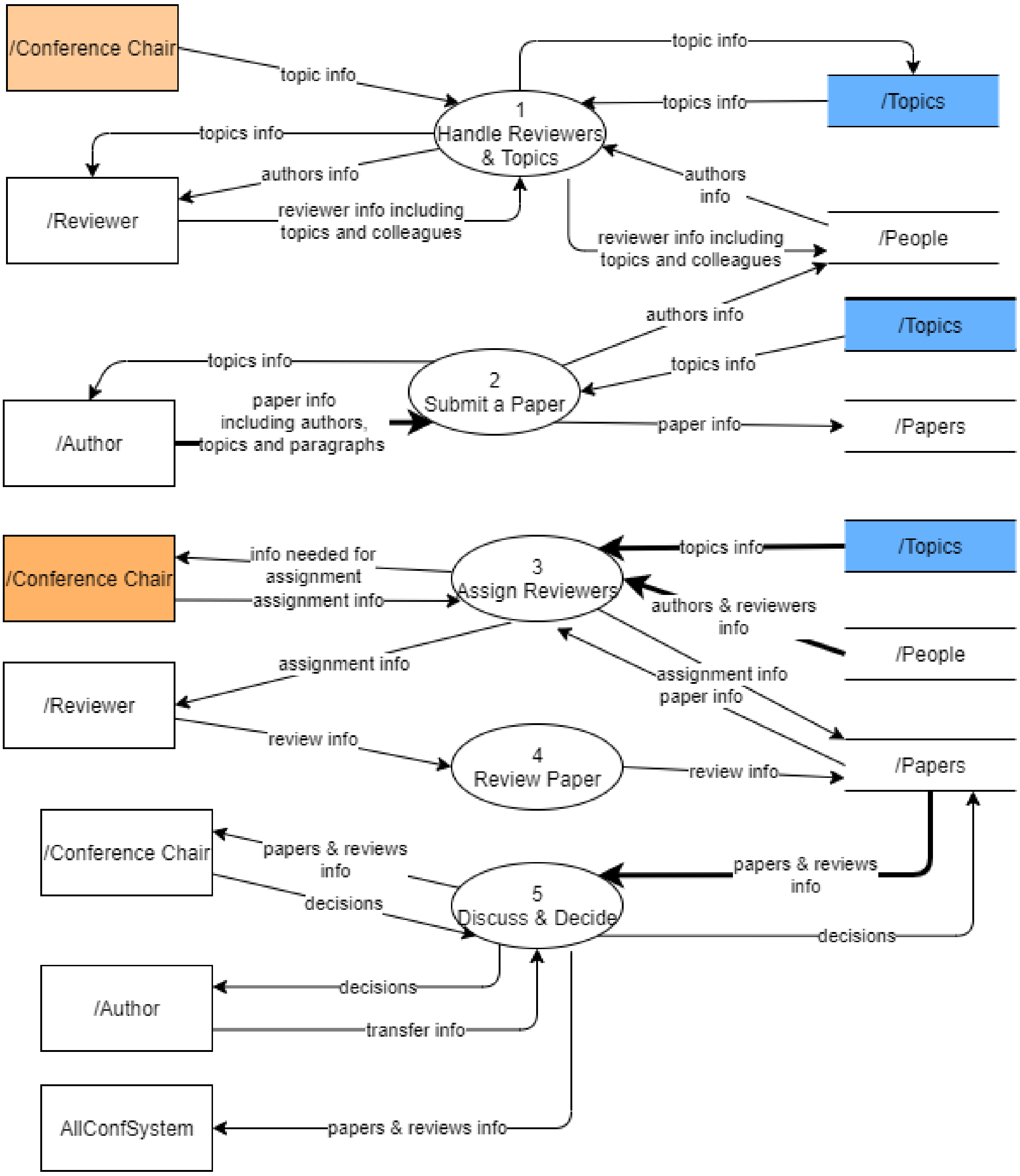
7. The reviewers' comments are for paragraphs in the papers, while a reviewer's score is for the entire paper. The information stored for a paragraph is its number in the paper, its title, the page on which it begins, and the page on which it ends.

8. The decisions are made by the conference chair, in accordance with the comments and scores. The authors are informed and details of rejected articles (including the reviewers' comments) are sent to the AllConfSystem system, after receiving permission from the authors.

1.2.3 Possible model of the new system (expected solution)

< Addition is orange, Modification in blue, use-as-is in black>



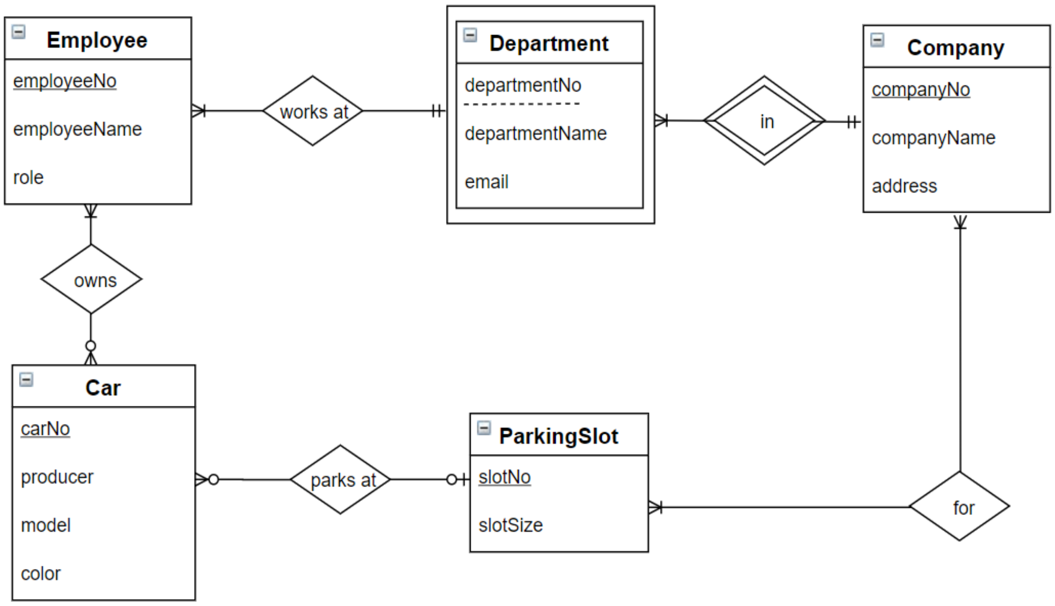

/Conference Chair
topic info
1
Handle Reviewers
& Topics
topic info
topics info
/Topics
topics info
/Reviewer
authors info
authors
info
reviewer info including
topics and colleagues
reviewer info including
topics and colleagues
/People
authors info
/Topics
topics info
topics info
2
Submit a Paper
/Author
paper info
including authors,
topics and paragraphs
paper info
/Papers
/Conference Chair
info needed for
assignment
assignment info
3
Assign Reviewers
topics info
/Topics
authors & reviewers
info
/People
/Reviewer
assignment info
assignment info
paper info
review info
4
Review Paper
review info
/Papers
/Conference Chair
papers & reviews
info
decisions
5
Discuss & Decide
papers & reviews
info
decisions
/Author
decisions
transfer info
AllConfSystem
papers & reviews info

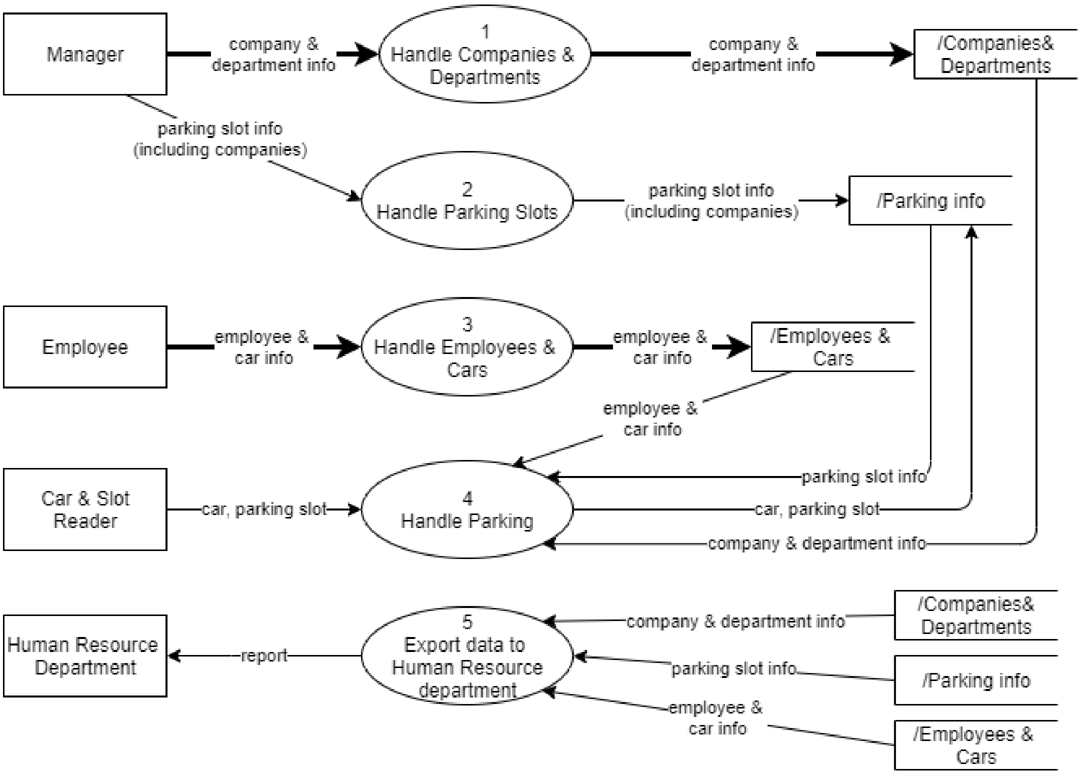
2 Experiment 2

2.1 Part 1

In this part, you will be required to model in ***ER and DF diagrams*** a system for ***performing assignments on departmental computers*** (hereinafter, *the new system*). For this purpose you can reuse parts of *an existing system* model for ***managing company's car parking***. Reuse can be done as-is, or with updates, additions or omissions. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements).

2.1.1 The model of the existing system





2.1.2 The requirements from the new system

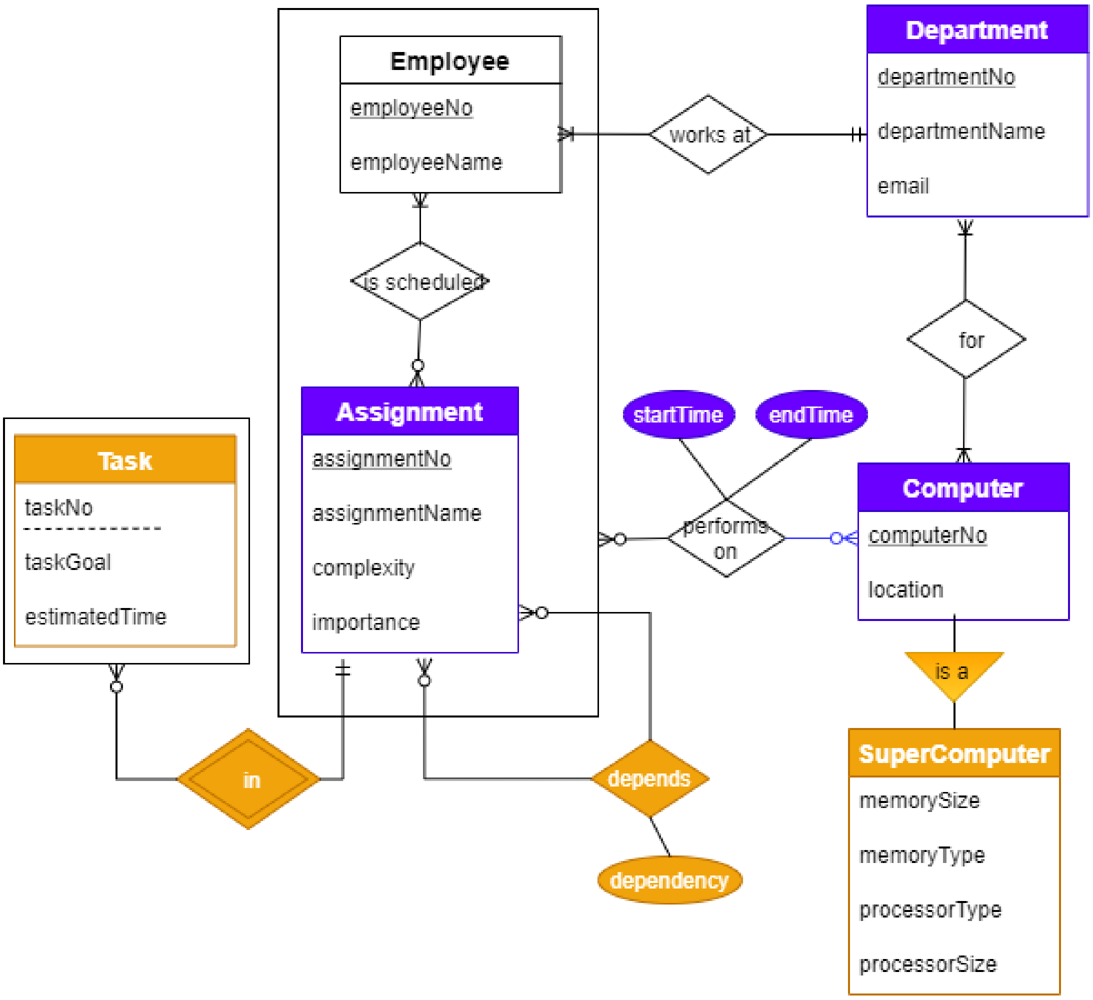
1. Each department in the company is assigned with computers that are capable of performing assignments for the employees of the departments to which they are associated.
2. For each employee, the following information is stored: employee number, name and the department in which he/she is employed. The department is characterized by its number, name and email address.
3. Employees are associated with assignments, where an employee can have multiple assignments associated at the same time. Each assignment has its name (which is not necessarily unique), its degree of complexity and its importance.
4. In addition, each assignment has its associated employees and the tasks from which it is composed (serial number of the task, its purpose and the amount of time estimated to achieve it). Dependencies are defined between assignments (immediately follows or eventually follows).
5. The computers, for which a unique number and physical place are saved, perform assignments for employees associated with those assignments. Execution times, including start and end times, are recorded. Some

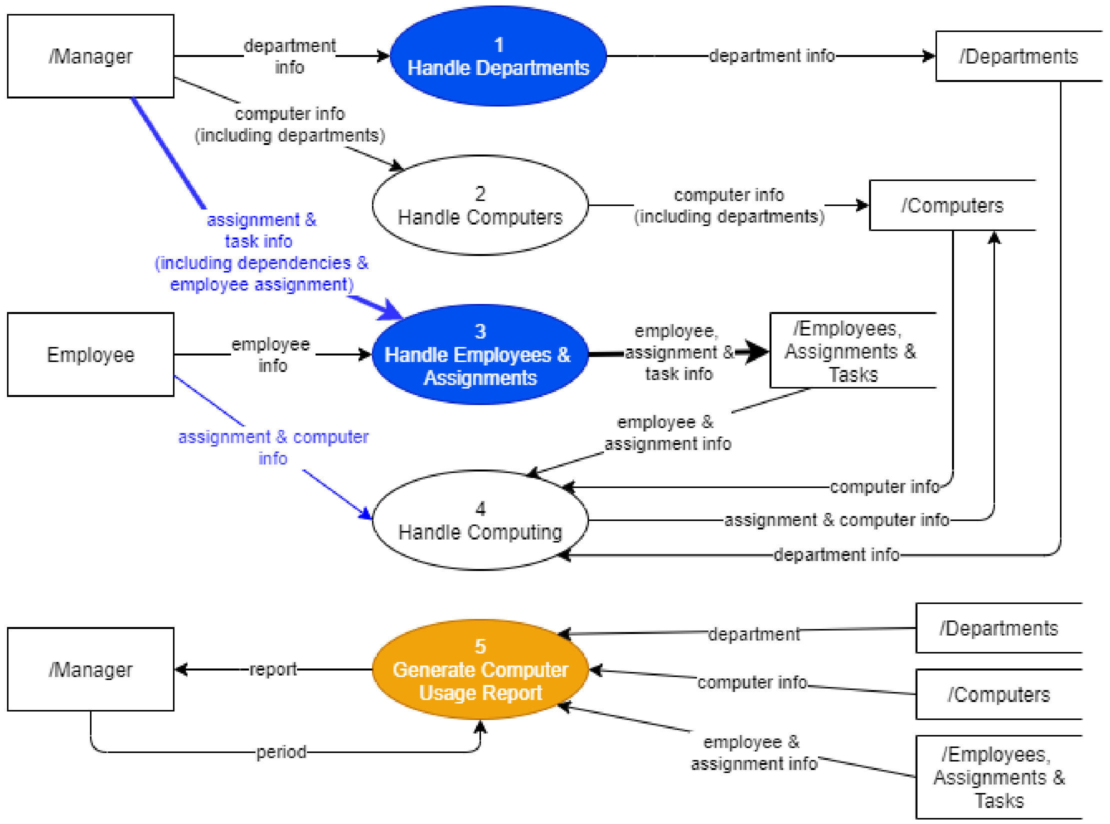
computers are "supercomputers" and have unique characteristics of memory type and size, as well as processor type and size.

6. The system supports managing the company departments by a manager, who is also responsible for defining all computers (including their association to departments), the assignments and their tasks, the dependencies among assignments and the assignment of employees for assignments.
7. When an employee requests to run an assignment under his/her responsibility on a computer, he/she provides the system with the assignment data and computer information. The system checks the data provided and allows the assignment to run on the computer if the employee is associated with the assignment and the computer is assigned to the employee’s department. If the assignment runs, its execution times will be saved.
8. The manager can generate a periodic report (the period is given as a parameter) that displays information about the degree of usage of computers for the various assignments by department.

2.1.3 Possible model of the new system (expected solution)

< Addition is orange, Modification in blue, use-as-is in black>

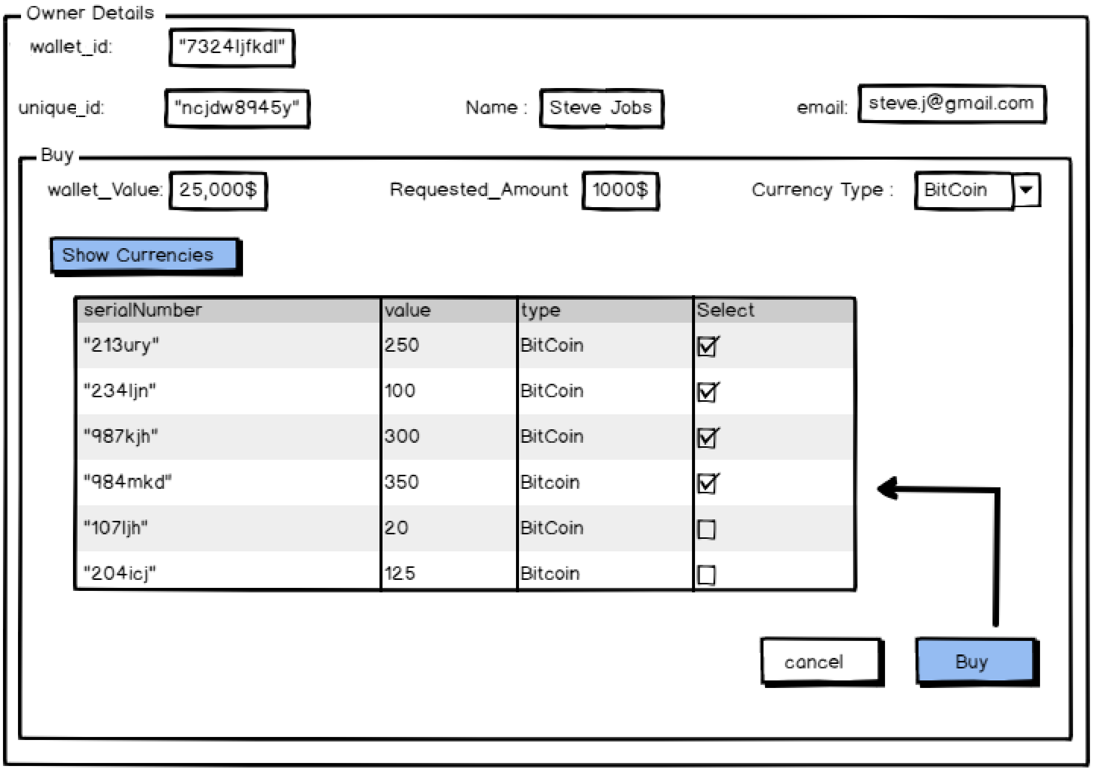



/Manager
department info
1
Handle Departments
department info
/Departments
computer info
(including departments)
2
Handle Computers
computer info
(including departments)
/Computers
assignment &
task info
(including dependencies &
employee assignment)
Employee
employee info
3
Handle Employees & Assignments
employee,
assignment &
task info
/Employees,
Assignments &
Tasks
assignment & computer info
employee &
assignment info
computer info
assignment & computer info
4
Handle Computing
department info
/Manager
report
5
Generate Computer Usage Report
department
/Departments
computer info
/Computers
employee &
assignment info
/Employees,
Assignments &
Tasks
period

2.2 Part 2

In this part, you will be required to model in ***UML use case and class diagrams*** the next generation of a ***digital wallet management system*** (hereinafter, *e-wallets2, the new system*). To do this, you can reuse parts of *an existing system* model for ***managing digital wallets – e-wallets1***. Reuse can be as-is, or with updates, additions or omissions. *Note: the existing system and the new one are modeled in different modeling languages*. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements) and that you have answered the requirements of each section (in the required language).

2.2.1 The model of the existing system

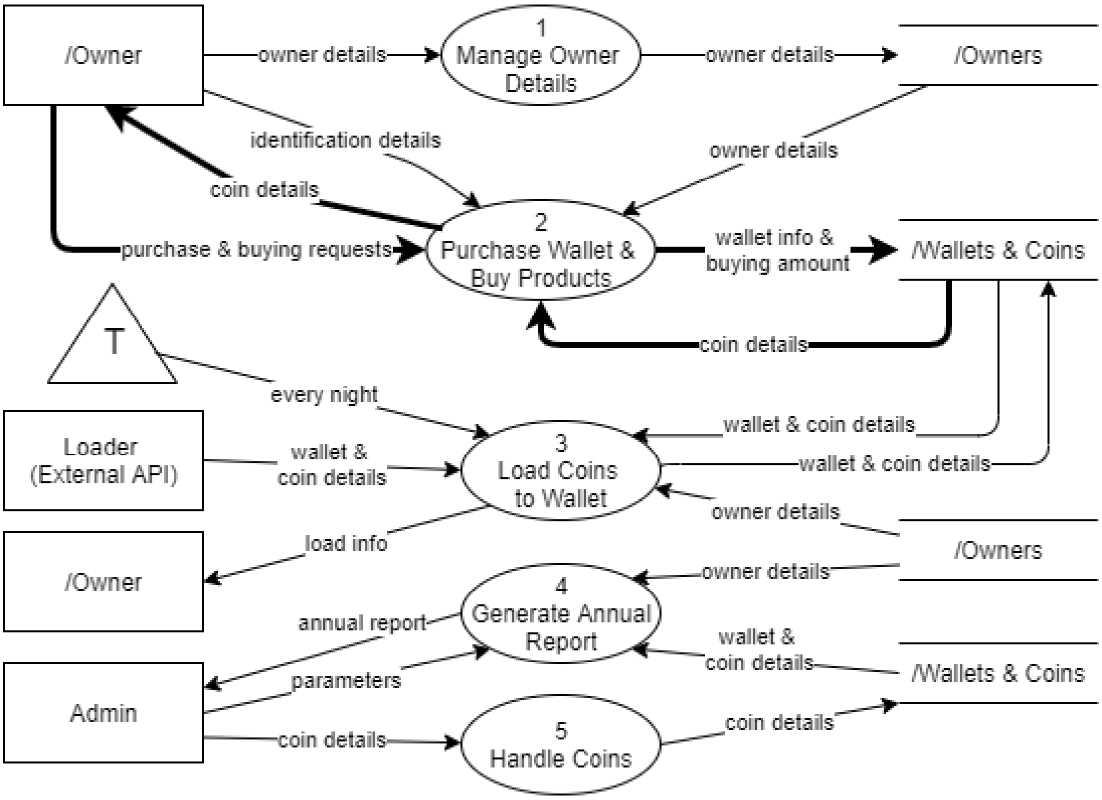


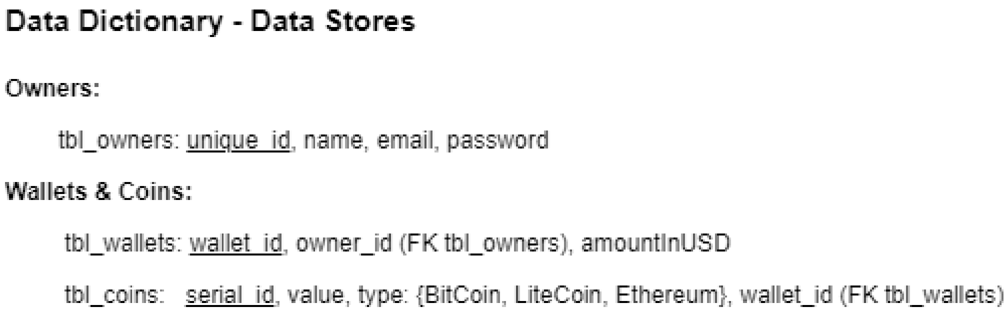
wallet_id: "7324ljfkdl"

unique_id: "ncjdw8945y" Name : Steve Jobs email: steve.j@gmail.com

wallet_Value: 25,000$ Requested_Amount 1000$ Currency Type : BitCoin

serialNumber	value	type	Select
"213ury"	250	BitCoin	☑
"234ljn"	100	BitCoin	☑
"987kjh"	300	BitCoin	☑
"984mkd"	350	Bitcoin	☑
"107ljh"	20	BitCoin	□
"204icj"	125	Bitcoin	□





Data Dictionary - Data Stores

Owners:

tbl_owners: unique_id, name, email, password

Wallets & Coins:

tbl_wallets: wallet_id, owner_id (FK tbl_owners), amountInUSD

tbl_coins: serial_id, value, type: {BitCoin, LiteCoin, Ethereum}, wallet_id (FK tbl_wallets)

2.2.2 The requirements from the new system

1. Similarly to the existing e-wallet1 system, the new e-wallet2 system manages digital wallets and enables the purchase of wallets, loading coins into them by the owner, and purchasing products that reduce the total value of the wallet.

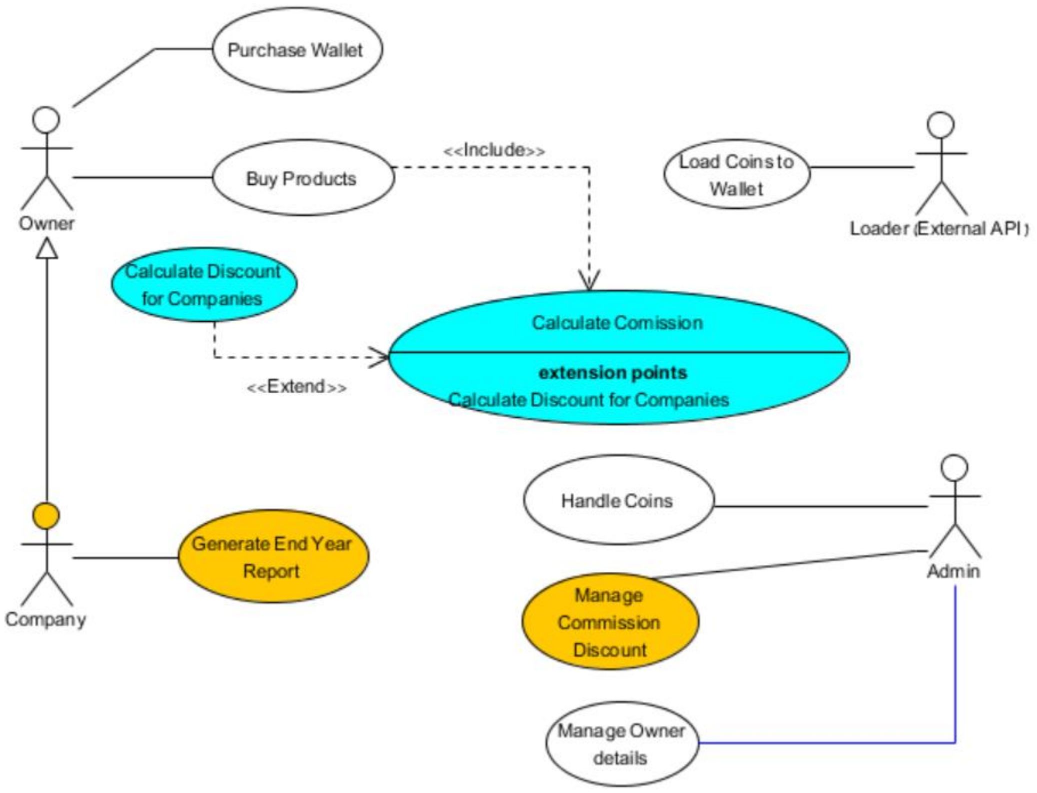
However, there are a number of changes to the new system. In particular, the e-wallet2 system supports the following options:

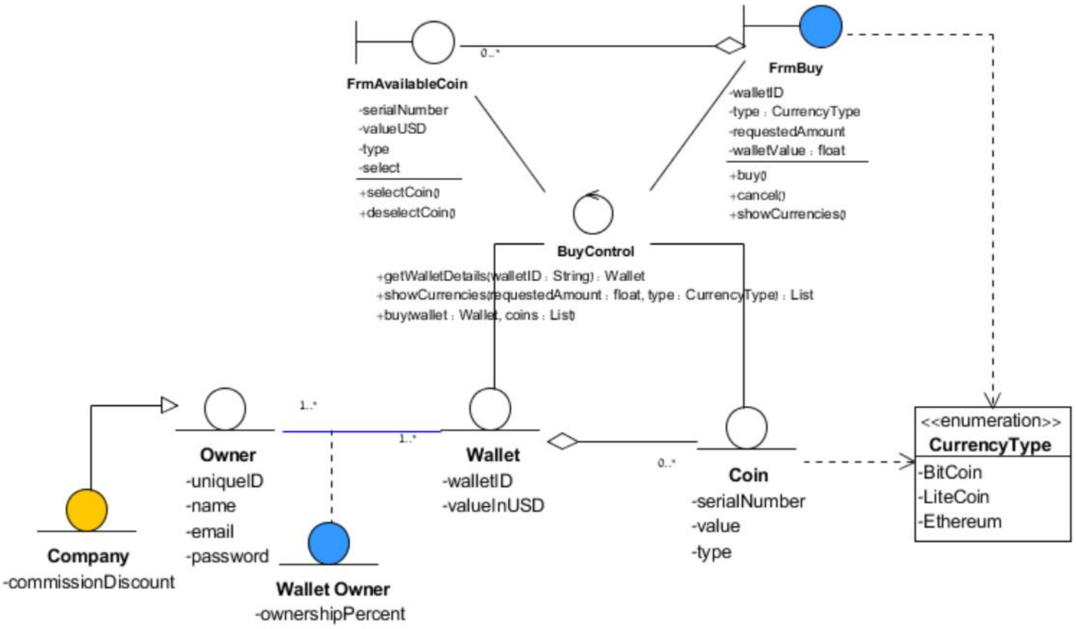
2. The information about the wallets, owners and their contents (i.e. coins) is the same as the information stored in the e-wallet1 system.

3. The information about the currencies themselves, including the supported types, is also the same.
4. A wallet can have multiple owners. The percentage of ownership of a customer on each of its wallets is kept in the system. The information about the system's customers (wallet owners) is managed by the admin.
5. When one buys products, the commission required for the operation (5% of the purchase value) is calculated.
6. Wallet owners can be individuals or companies. For a company, the amount of the commission discount is also maintained, a figure updated by the admin from time to time. When calculating the commission for buying products, this discount is taken into account to the extent that a majority (more than 50%) of wallet’s ownership is that of companies.
7. Every year, each company receives a summary report on the wallets it owns, the ownership percentages and their contents (i.e. the currencies in them).
8. For security reasons when purchasing products, no wallet owner information is displayed. After identification (in standard ways and therefore there is no need to model the identification), the user is required to enter the ID of the wallet, the amount (in USD) and the requested currency. When he/she clicks “show currency” (provided that the user is the owner of the wallet), the total amount is displayed (in USD) as well as the coins in the wallet that can be used to buy products (by marking the relevant currencies and clicking buy, as in the e-wallet1 system).

2.2.3 Possible model of the new system (expected solution)

<Addition is orange, Modification in blue, use-as-is in black>



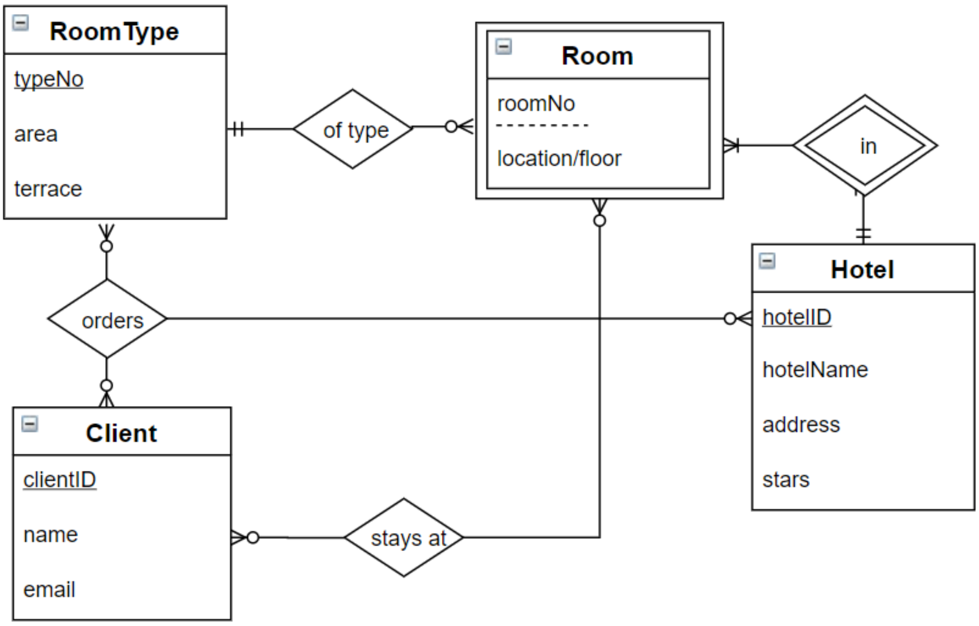


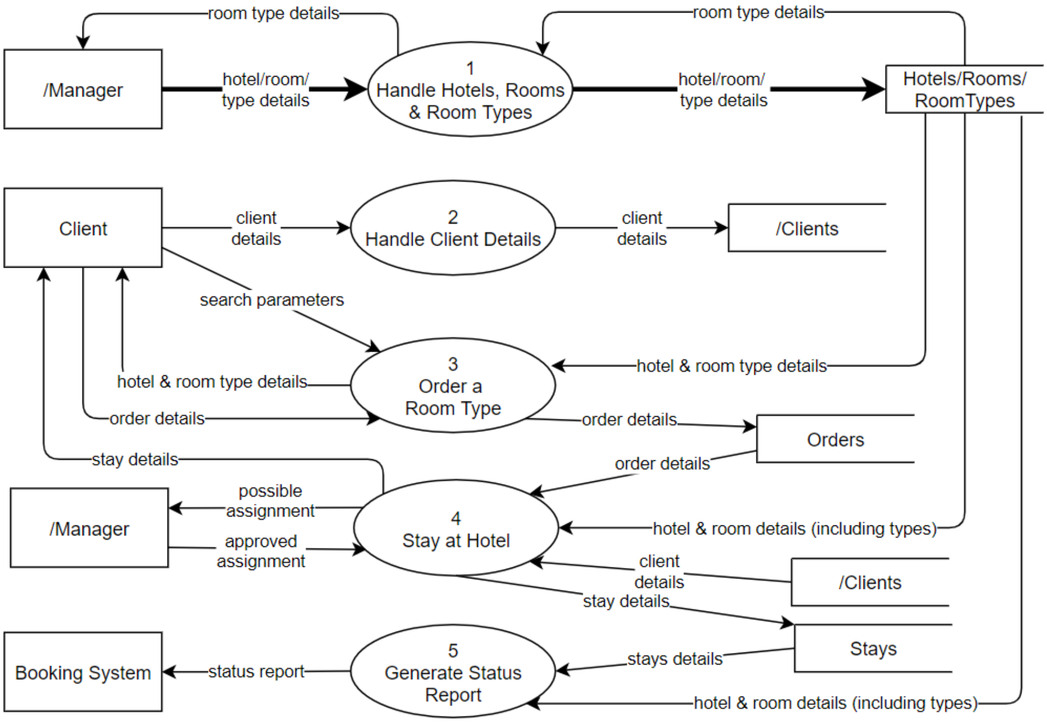
3 Experiment 3

3.1 Part 1

In this part, you will be required to model in ***ER and DF diagrams*** a system for ***managing files and changes*** (hereinafter, *the new system*). For this purpose you can reuse parts of *an existing system* model of ***hotel management***. Reuse can be done as-is, or with updates, additions or omissions. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements).

3.1.1 The model of the existing system



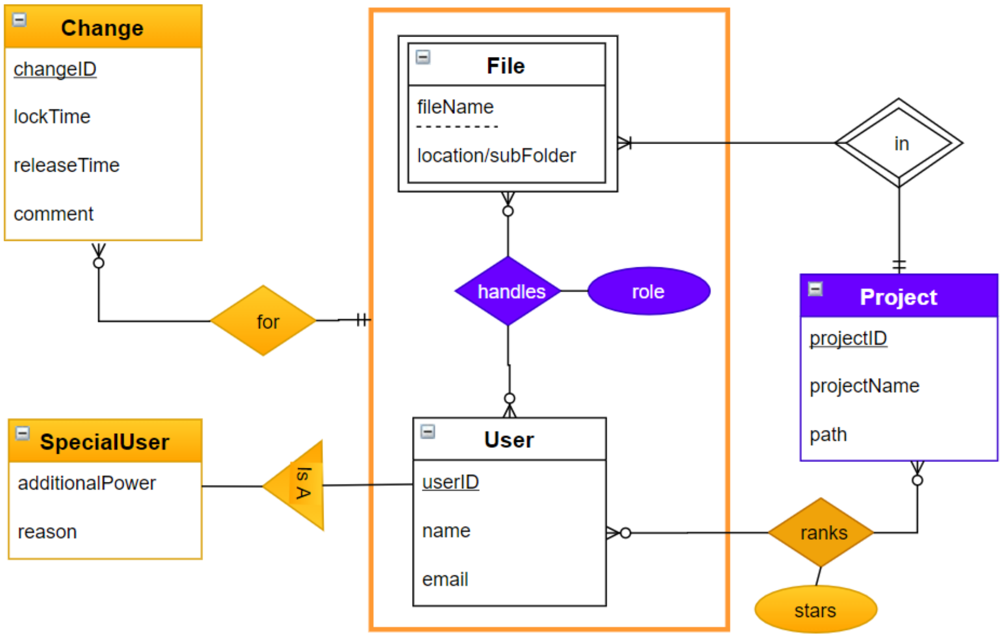


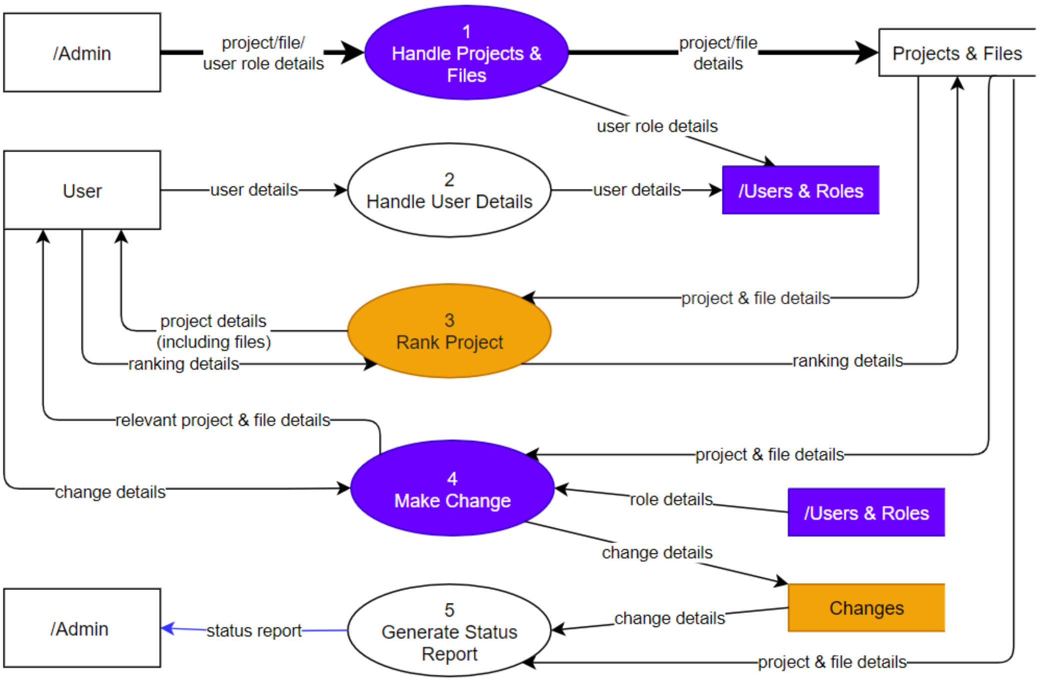
3.1.2 The requirements from the new system

1. The system supports the management of projects and files by the administrator, including defining roles for the different users for each file.
2. The system also enables the management of the information of all users.
3. A user can evaluate projects (by providing stars) based on the information available about them and their files.
4. A user can also make a change to files—in this case, he/she is presented with files and can be modified (depending on the roles that are set and the status of locking the files by other users).
5. The manager can produce a status report that displays information about the changes made to the project files in the past month.

3.1.3 Possible model of the new system (expected solution)

<Addition is orange, Modification in blue, use-as-is in black>

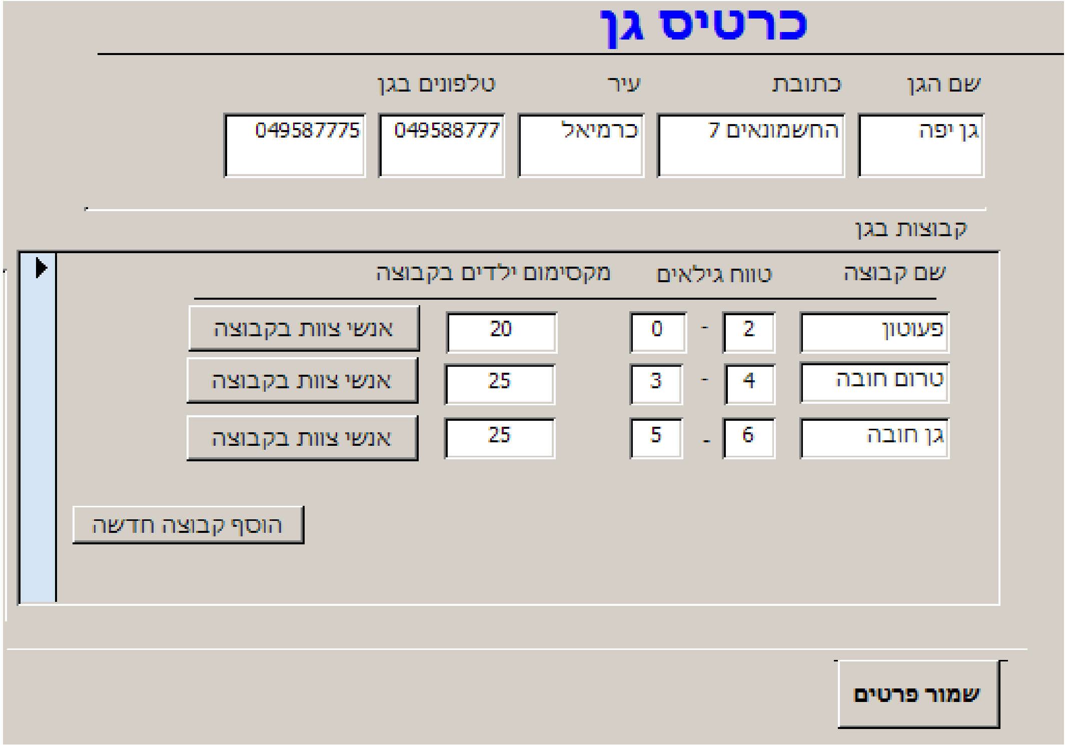




3.2 Part 2

In this part, you will be required to model in ***UML use case and class diagrams*** the next generation of a ***kindergarten network management system*** *(hereinafter, kindergarten2, the new system*). To do this, you can reuse parts of *an existing system* model for ***managing kindergarten*** – ***kindergarten1***. Reuse can be as-is, or with updates, additions or omissions. *Note: the existing system and the new one are modeled in different modeling languages*. The evaluation of each section will be according to two metrics: (1) the degree to which the requirements of the new system are met, and (2) the degree of reuse of the existing system model. Make sure that there is no redundancy (unnecessary elements) and deficiencies (missing elements) and that you have answered the requirements of each section (in the required language).

3.2.1 The model of the existing system

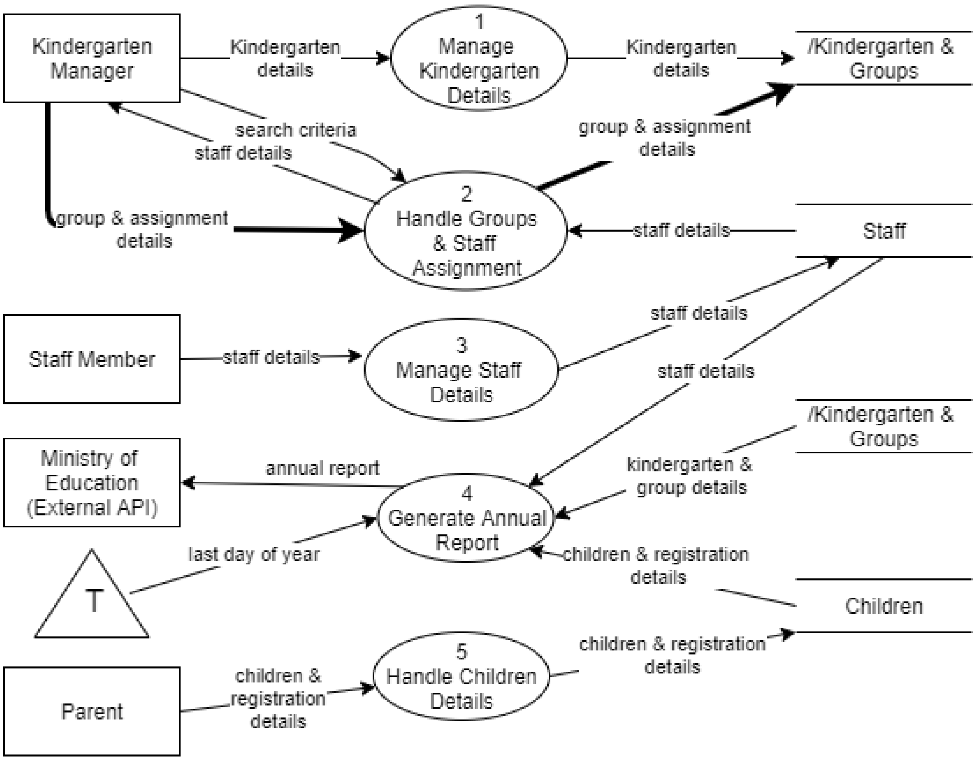


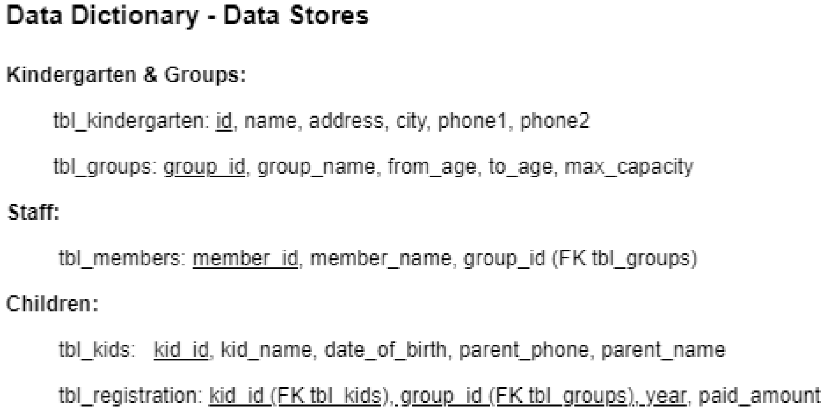
כרטיס גן

שם הגן	כתובת	עיר	טלפונים בגן	
גן יפה	החשמונאים 7	כרמיאל	049588777	049587775

קבוצות בגן

שם קבוצה	טווח גילאים	מקסימום ילדים בקבוצה
פעוטון	0 - 2	20
טרום חובה	3 - 4	25
גן חובה	5 - 6	25


Kindergarten Manager
Kindergarten details
1 Manage Kindergarten Details
Kindergarten details
/Kindergarten & Groups
search criteria
staff details
group & assignment details
2 Handle Groups & Staff Assignment
group & assignment details
staff details
Staff
Staff Member
staff details
3 Manage Staff Details
staff details
staff details
Ministry of Education (External API)
annual report
/Kindergarten & Groups
kindergarten & group details
4 Generate Annual Report
last day of year
T
children & registration details
Children
children & registration details
Parent
children & registration details
5 Handle Children Details



Data Dictionary - Data Stores

Kindergarten & Groups:

tbl_kindergarten: id, name, address, city, phone1, phone2

tbl_groups: group_id, group_name, from_age, to_age, max_capacity

Staff:

tbl_members: member_id, member_name, group_id (FK tbl_groups)

Children:

tbl_kids: kid_id, kid_name, date_of_birth, parent_phone, parent_name

tbl_registration: kid_id (FK tbl_kids), group_id (FK tbl_groups), year, paid_amount

3.2.2 The requirements from the new system

The new kindergarten2 system manages information about a network of kindergartens, including their groups and staff members. In particular, the system supports the following functionality (only):

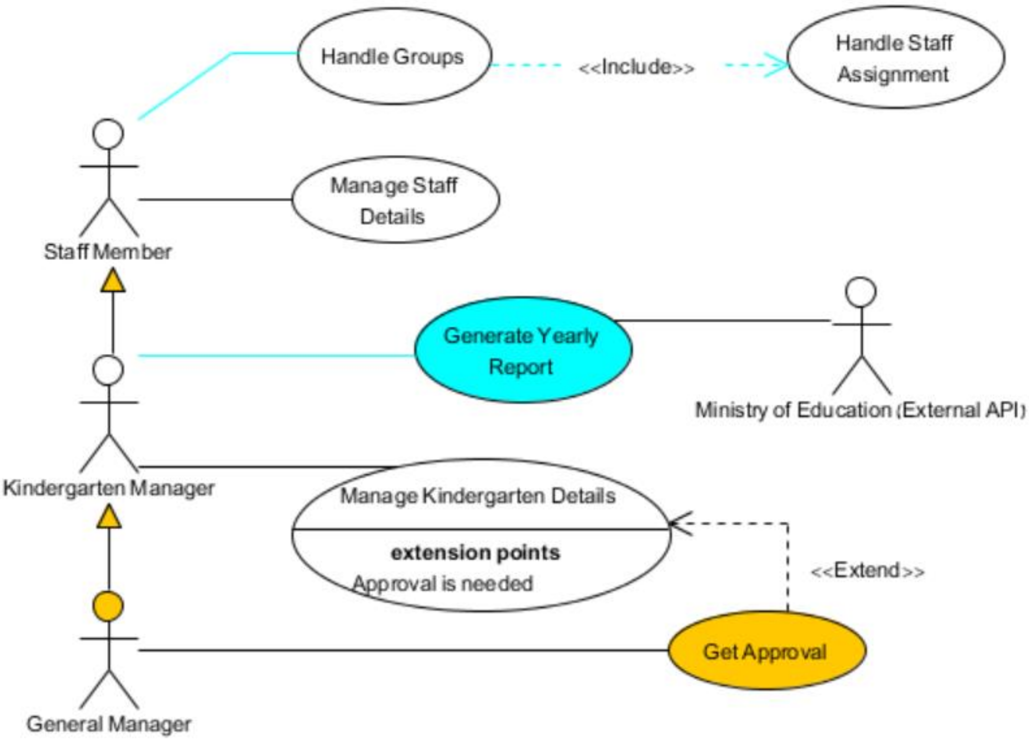
1. Similarly to the existing system, in the new system, team members manage the information about themselves and the garden managers (who are team members themselves) manage information about the kindergartens themselves.
2. The information about staff members and kindergartens in both systems is similar, but instead of contact details (address, city and phones), the

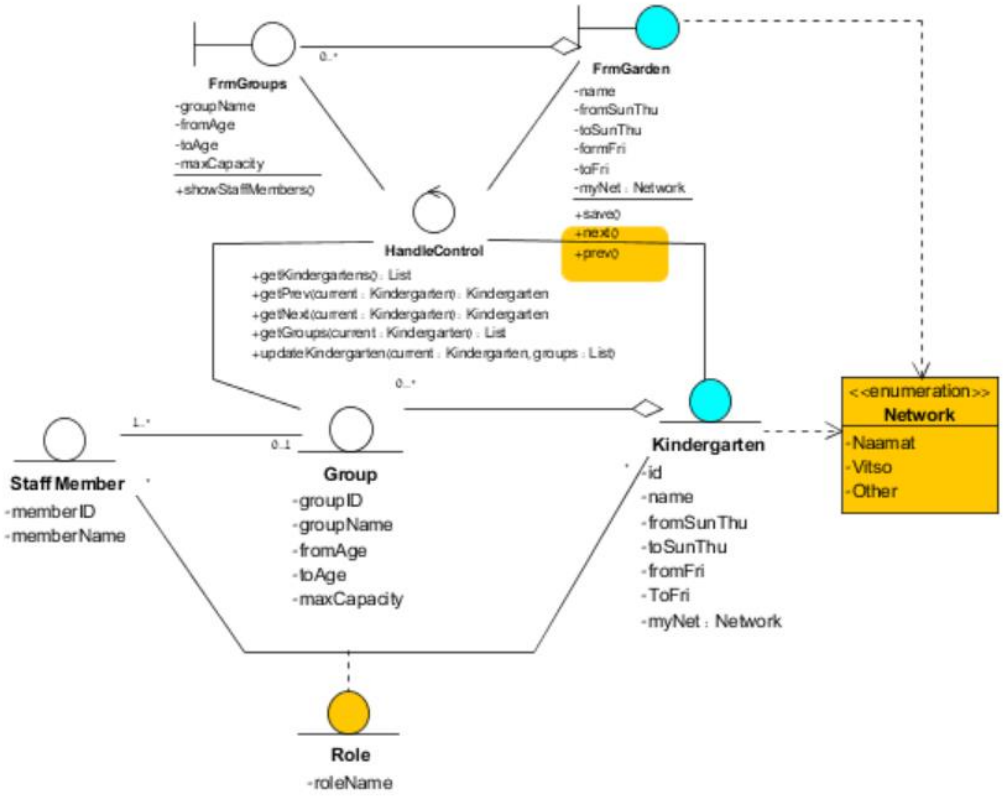
new system reserves for each kindergarden the range of hours it is open on weekdays (e.g. 7:00-17:00) and the range of operation hours on Fridays (e.g. 7:00-13:00). In addition, each kindergarten belongs to a network (Naamat, Wizo, or else).

3. The management of the groups, which also includes assignment of team members, is done by the team members themselves. In a kindergarten there can be several groups, each of which belongs exclusively to one kindergarten. The information about the groups on both systems is the same.
4. A team member can be associated with a specific group, or generally with a network — in the latter case the system presents in which kindergarten they work and what role in each kindergarten.
5. When updating the kindergarden information, if it is a large kindergarden (in terms of number of groups and/or total capacity of groups), approval from the network manager himself/herself, which is a kindergarten manager, is required.
6. The annual report, which is produced on the last day of the year and includes information about all kindergartens, including their groups and team members, is sent to both kindergarten managers and the Ministry of Education (through an External API, in order to publish and market the various kindergartens among interested parents).
7. The kindergarden information management screen in the new system is similar to the appropriate screen in the existing system: each kindergarten displays its details (depending on the data saved in the new system) at the top of the screen and its groups – at the center part of the screen; One can save details about a kindergarten and browse through the kindergartens (next, previous); One can edit the group information and view their team members (on a separate screen), but addition of new groups to the kindergarten is not supported on that screen.

3.2.3 Possible model of the new system (expected solution)

< Addition is orange, Modification in blue, use-as-is in black>




FrmGroups
-groupName
-fromAge
-toAge
-maxCapacity
+showStaffMembers()
0..*
FrmGarden
-name
-fromSunThu
-toSunThu
-formFri
-toFri
-myNet : Network
+save()
+next()
+prev()
HandleControl
+getKindergartens() : List
+getPrev(current : Kindergarten) : Kindergarten
+getNext(current : Kindergarten) : Kindergarten
+getGroups(current : Kindergarten) : List
+updateKindergarten(current : Kindergarten, groups : List)
0..*
<<enumeration>>
Network
-Naamat
-Vitso
-Other
1..*
0..1
Staff Member
-memberID
-memberName
Group
-groupID
-groupName
-fromAge
-toAge
-maxCapacity
Kindergarten
-id
-name
-fromSunThu
-toSunThu
-fromFri
-ToFri
-myNet : Network
Role
-roleName