Life Simulation Design Presentation

Software Engineering CENG 3004,
Course Project

Beyza Kurt - beyzakurt1998@hotmail.com
Mustafa Sakallı - mustafasakalli42@gmail.com
Dilek Koçak - dileekkocakk@gmail.com
Ala Şakko - lulu.shakko@gmail.com

OVERVIEW

Our application aims to perform one of the most difficult tasks in the world:
To give users a good time.

With our application, users can create their own players and families,
manage them in real-time, and perhaps even build their dream life.

Now our application has ***single-player usage***. But in the future, our users will be able to visit each other's homes online and their players can interact with each other.
We will even have mini-games in our app.

Players have 8 kinds of needs: **energy**, **hunger**, **bladder**, **hygiene**, **sociality**, and **fun**.

They can meet their basic needs at home such as sleeping, going to the toilet, taking a shower, eating, and chatting with housemates.

At the same time, players can play games, watch movies and do research on the computer; can read fiction, non-fiction or educational books; or they can do house hobbies. Players can also *go out*; can go to work, shopping and travel. Of course, this fun job has its hard sides. For example, if the user leaves their player hungry for 32 hours, I'm sorry they will learn that their player is dead the next time.

DESIGN GOALS

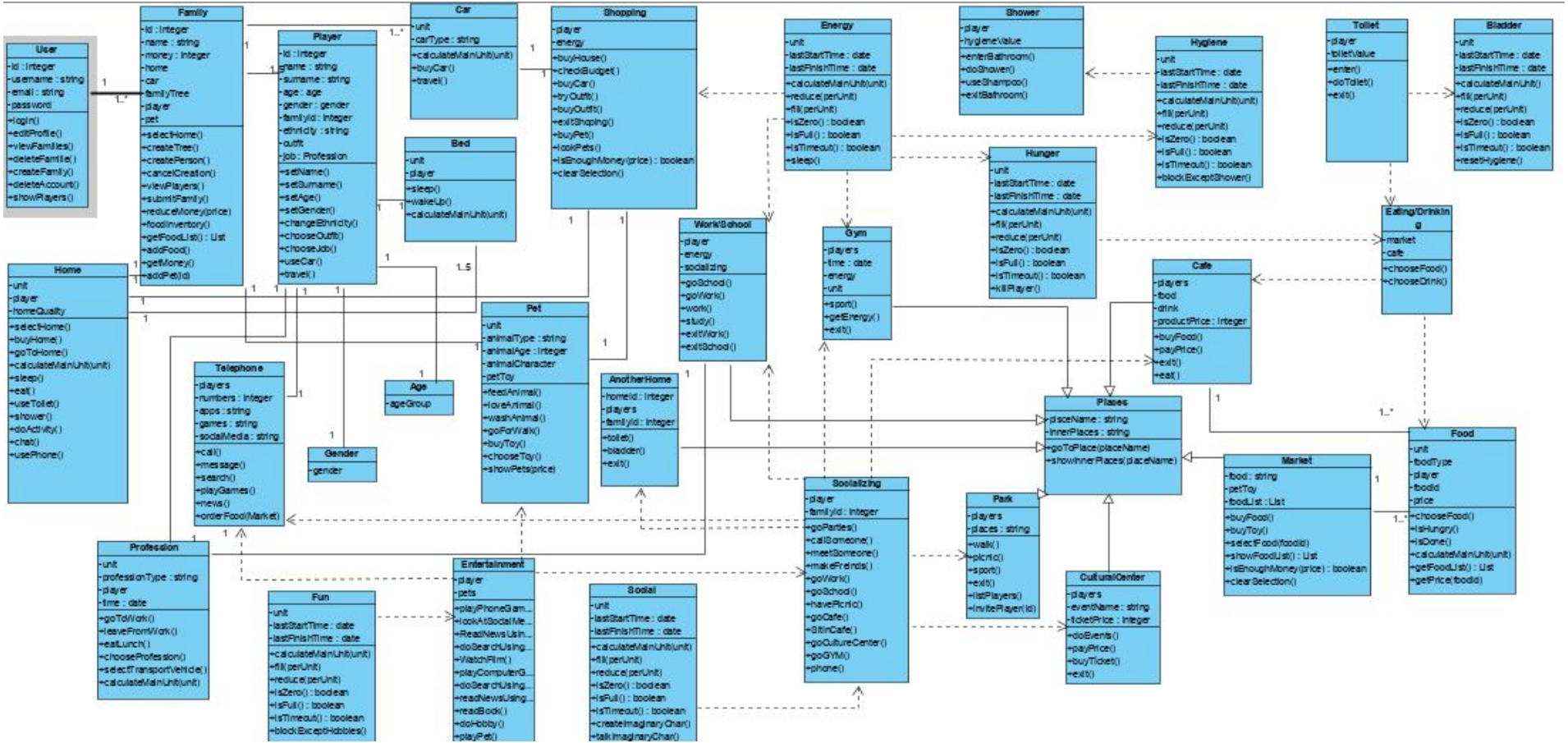
Goal's Concern	Related Requirement Identifier	Description
Reliability	NR-1	The application checks the validity of the mailing address entered while the User is being a member.
Security	NR-2	The user's password must be at least 8 and at most 16 characters and must contain at least one letter, at least one number, and at least one special character.
Data integrity	NR-4	The application checks e-mail address and username in the database, if these matches ask for different login
Recoverability	NR-5	The application sends a password reset email to the user if the user click the "Forgot Password" button

Portability	NR-6	The application runs on Android and IOS platforms.
efficiency	NR-7	A website should be capable enough to handle 1000 users without affecting its performance.
Operability	NR-8	The application requires an internet connection to work
Timeliness	NR-9	The application clock runs simultaneously with real timezone.
Accessibility	NR-10	Each family consists of at least 1 and at most 5 players.
Relevance	NR-11	When the family creation is completed, the application automatically determines the family fund. The family fund is determined for how many players the family holds, with $ 2000 per player -> 2000 x (number of players) = family accumulation
Capacity	NR-12	A User can have up to 5 families.
Capacity	NR-13	Each player's inventory capacity is 5 slots.
Relevance	NR-14	The Player returns to the home screen as a result of the User canceling the action of the Player.

SYSTEM MODELS

- Class Diagram
- Sequence Diagrams
  - Order Food with Mobilephone
  - Eat at the Home
  - Go to the Picnic
  - Buy a Pet
- Activity Diagrams
  - Create a Family
  - Chat with another Player
  - Eat at the Home
  - Go to the Theatre
- Statechart Diagram

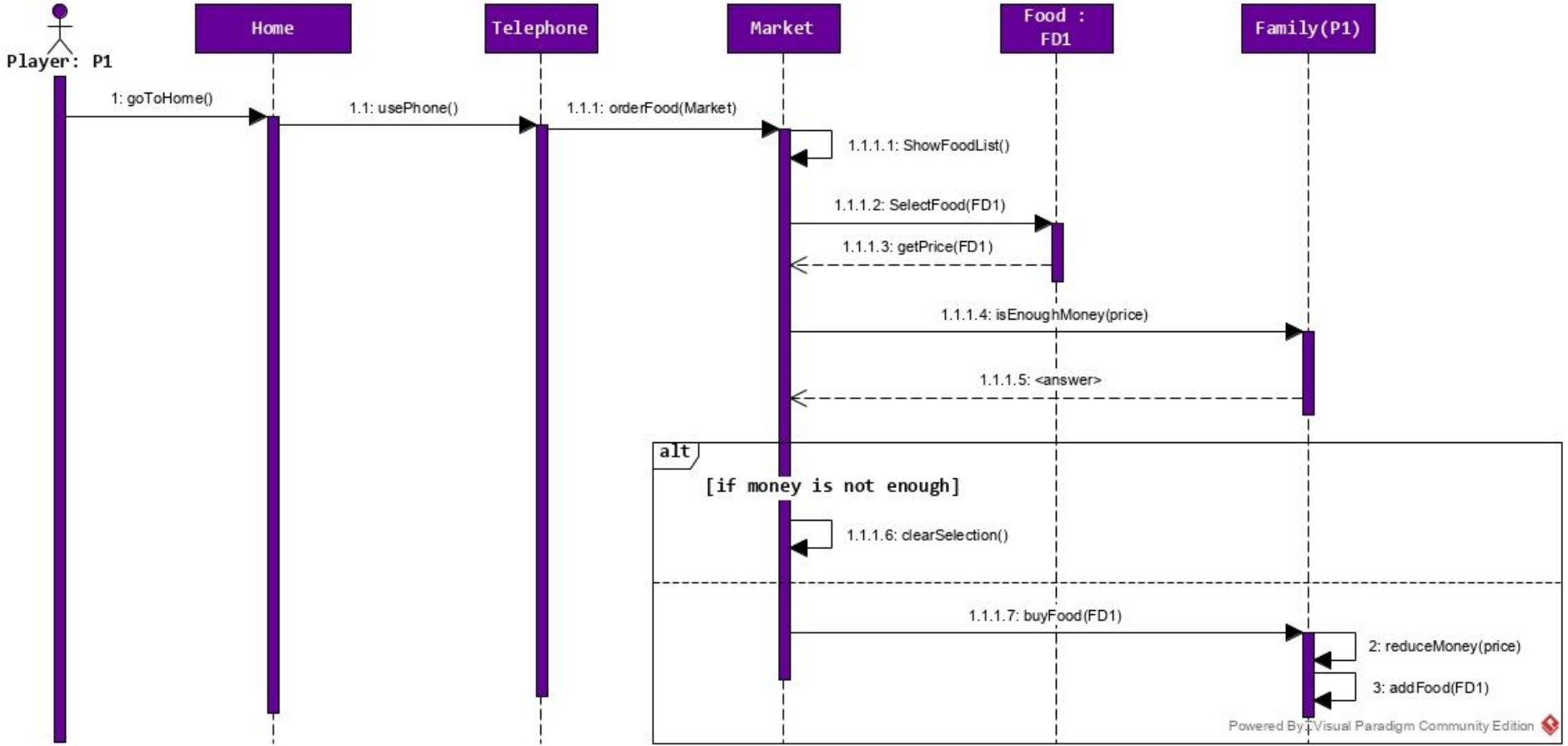
Class Diagram



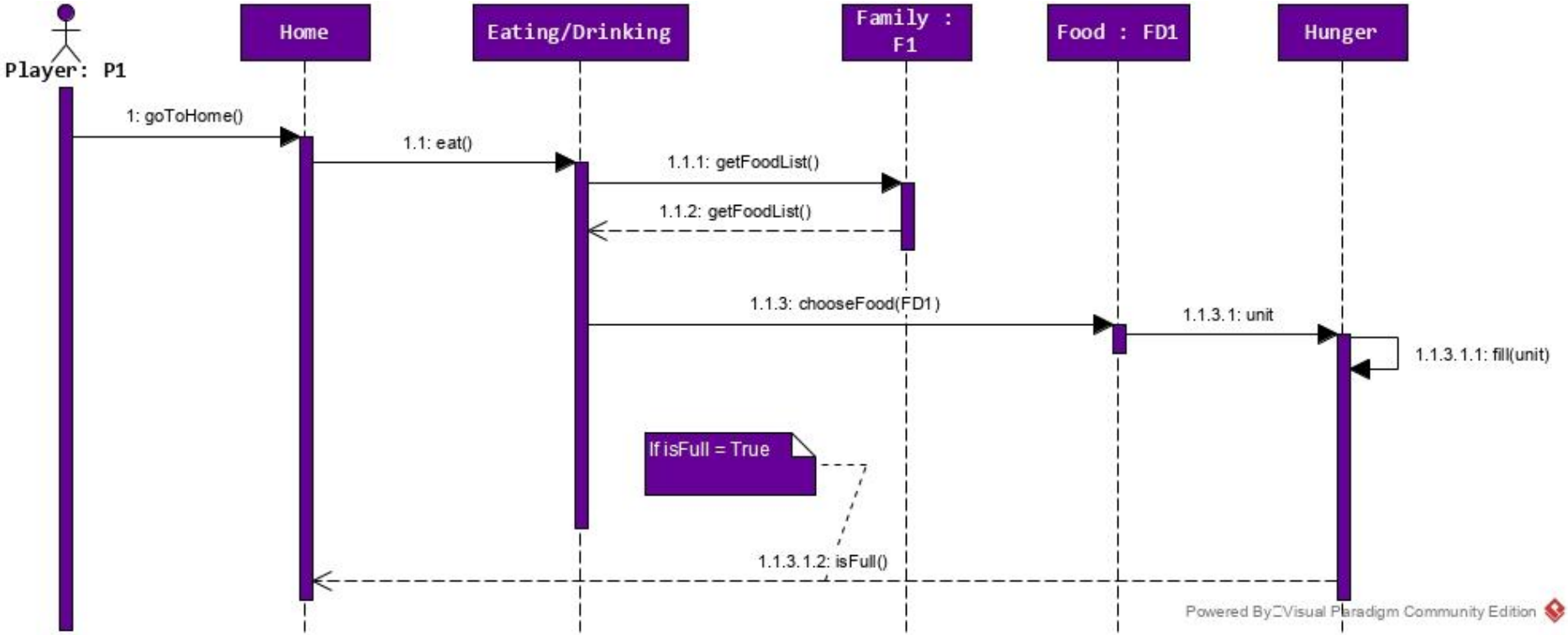
Sequence Diagrams

1. Order Food with Mobilephone
2. Eat at the Home
3. Go to the Picnic
4. Buy a Pet

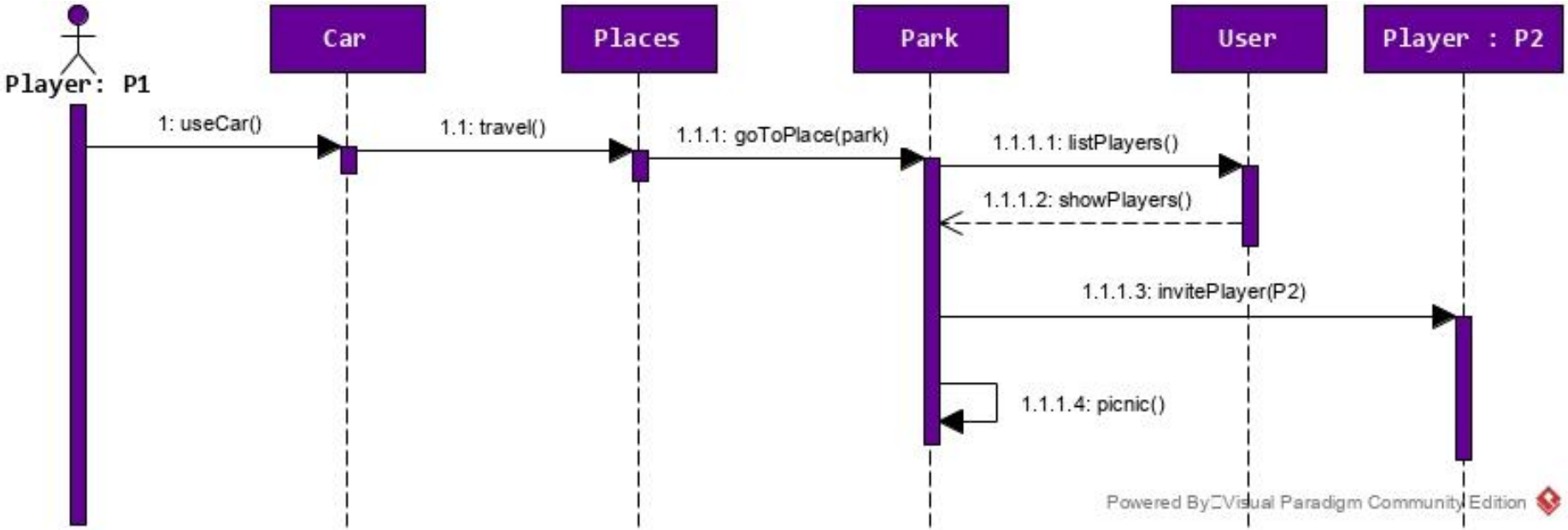
1. Order Food with Mobilephone



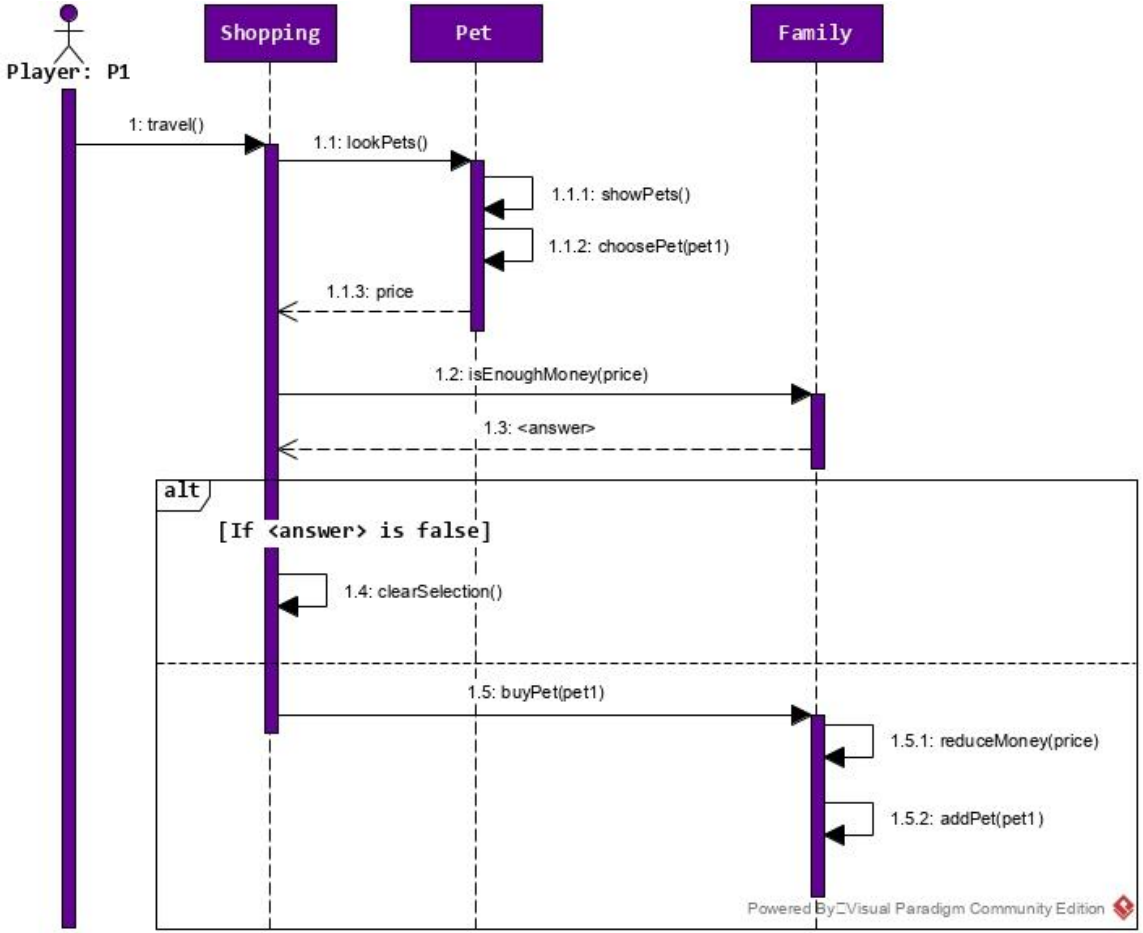
2. Eat at the Home



3. Go to the Picnic



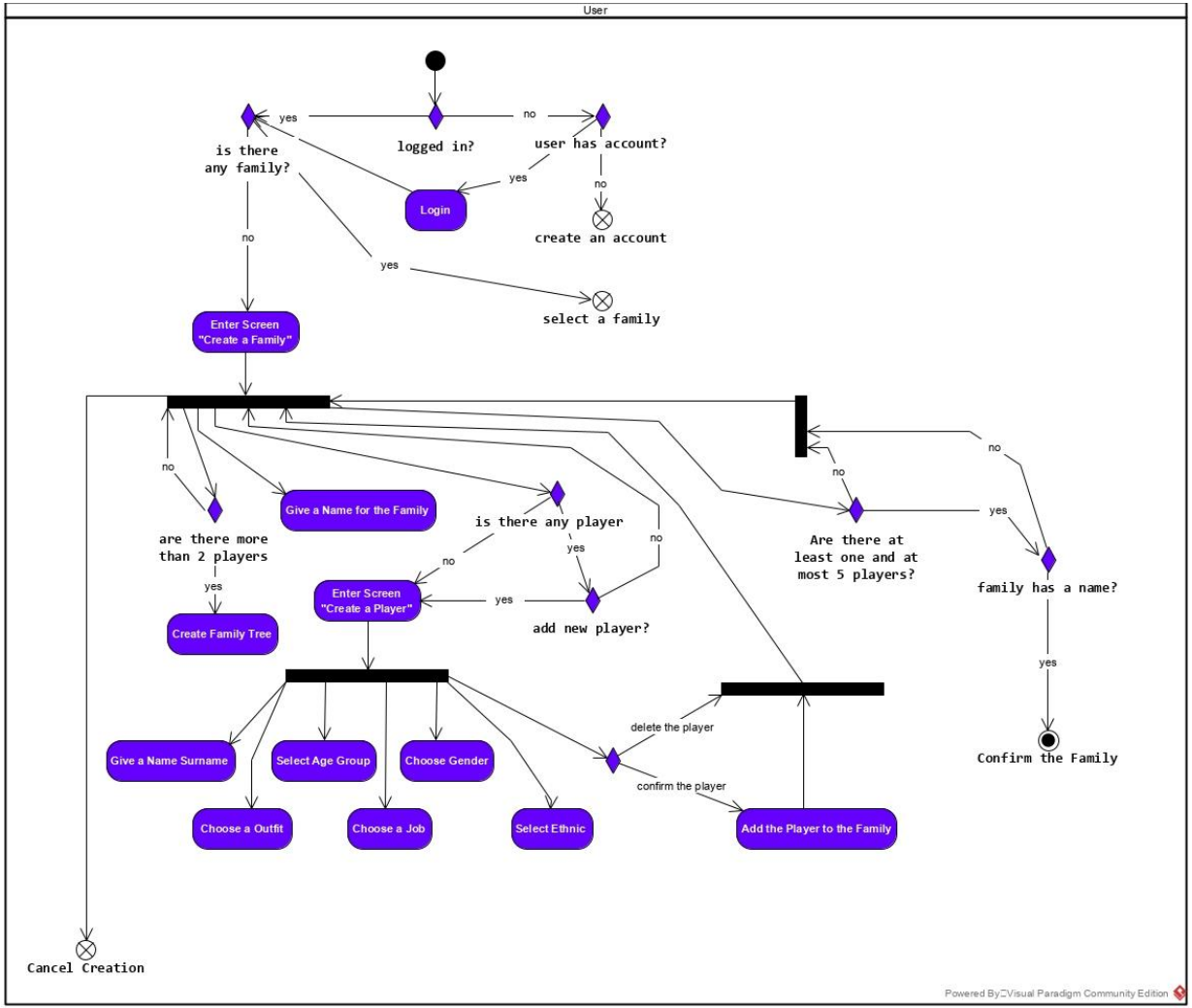
4. Buy a Pet


Player: P1
Shopping
Pet
Family
1: travel()
1.1: lookPets()
1.1.1: showPets()
1.1.2: choosePet(pet1)
1.1.3: price
1.2: isEnoughMoney(price)
1.3: <answer>
alt
[If <answer> is false]
1.4: clearSelection()
1.5: buyPet(pet1)
1.5.1: reduceMoney(price)
1.5.2: addPet(pet1)
Powered By Visual Paradigm Community Edition

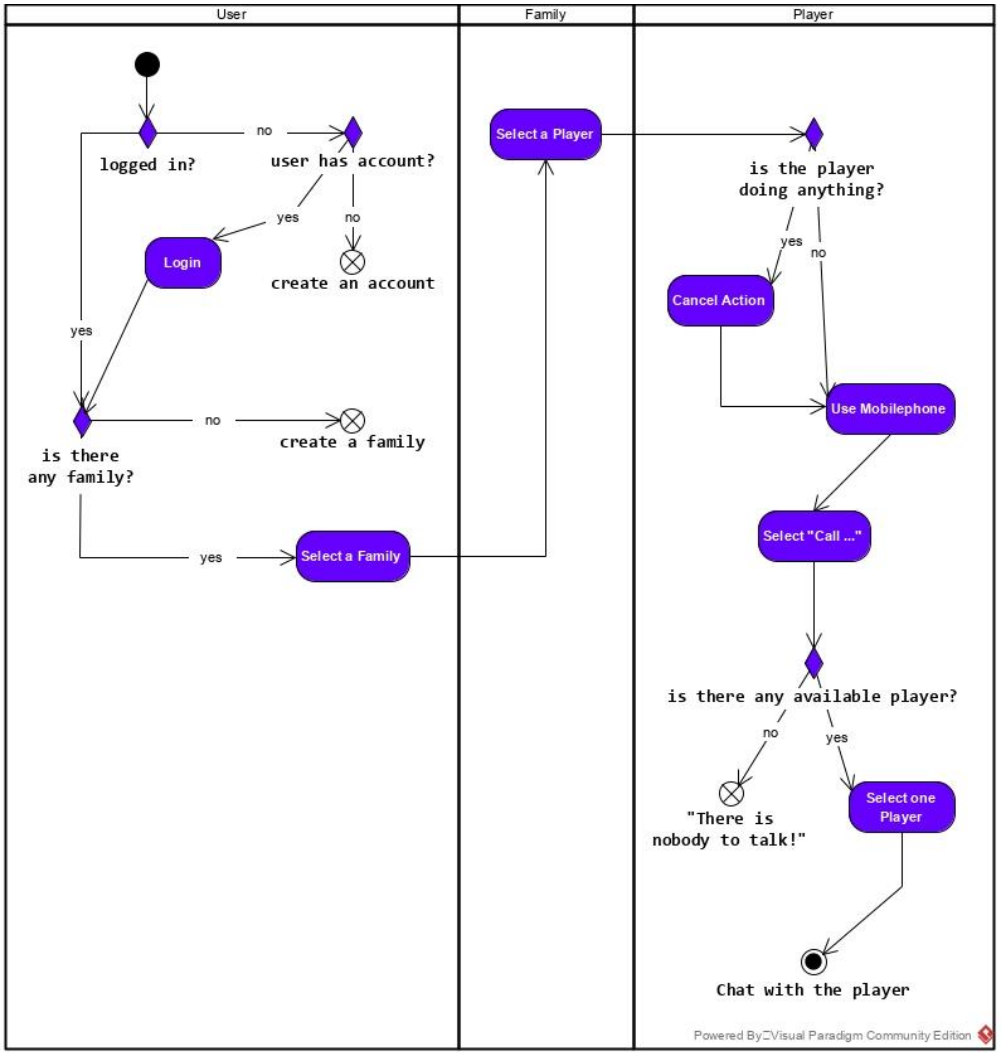
Activity Diagrams

1. Create a Family
2. Chat with another Player
3. Eat at the Home
4. Go to the Theatre

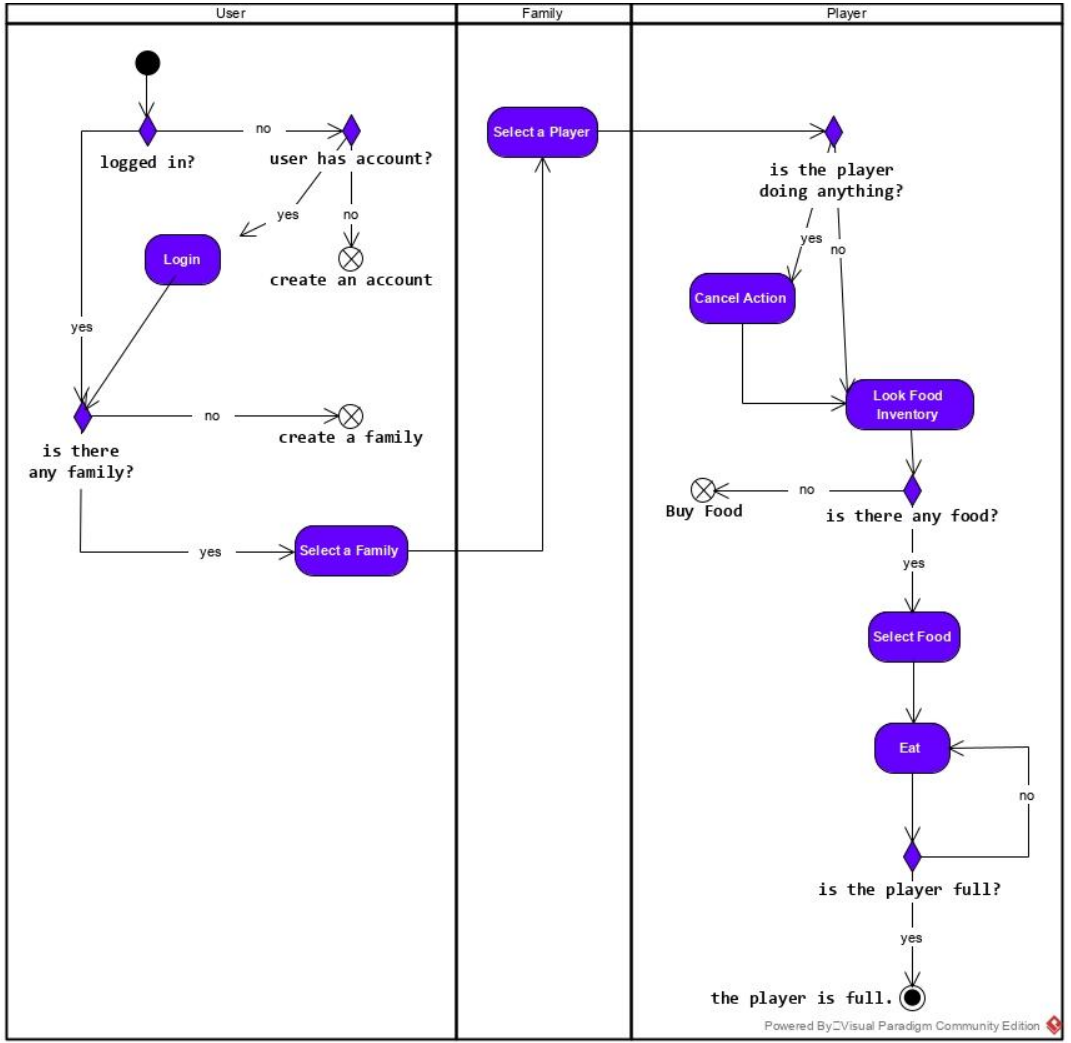
1. Create a Family



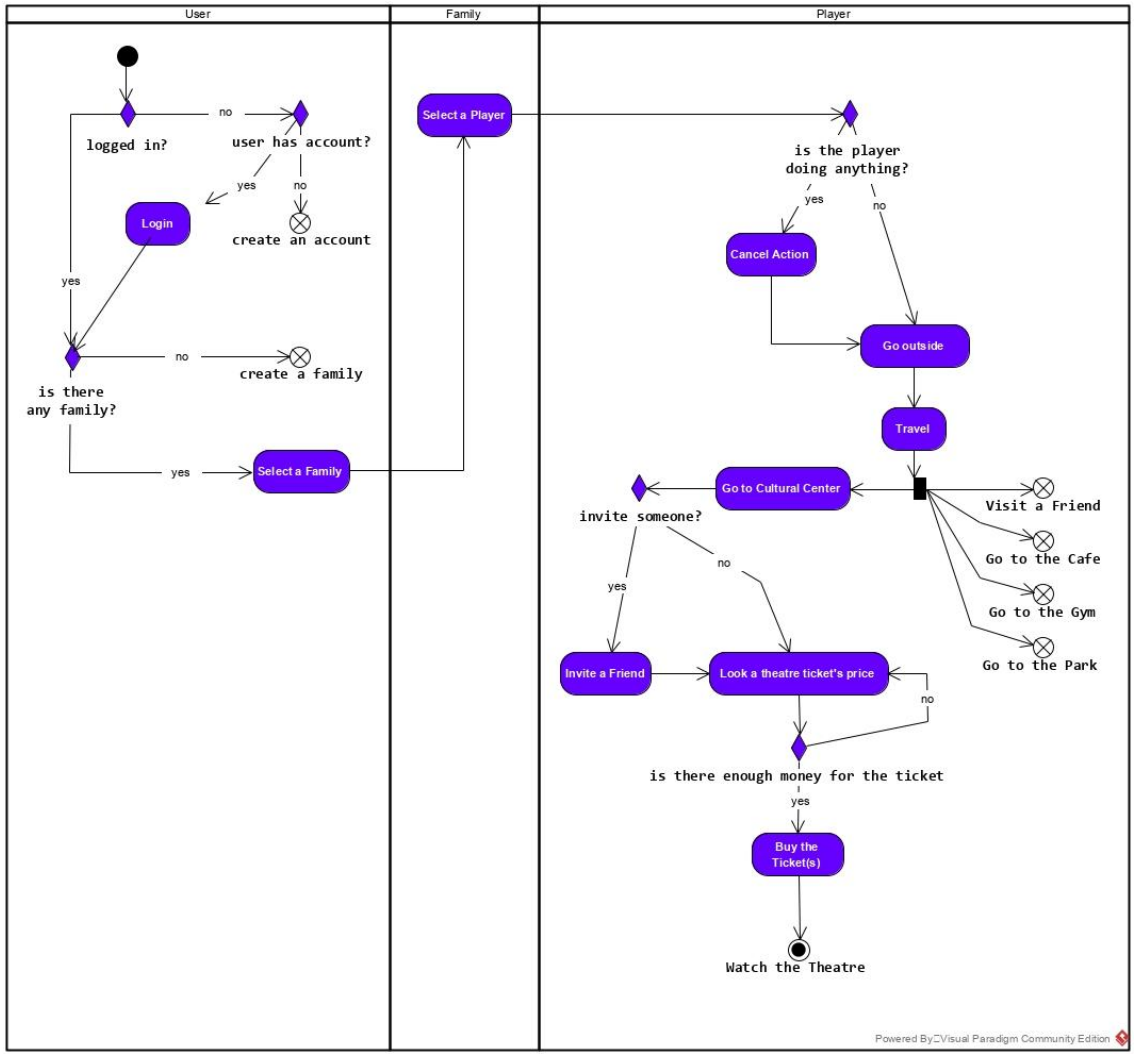
2. Chat with another Player



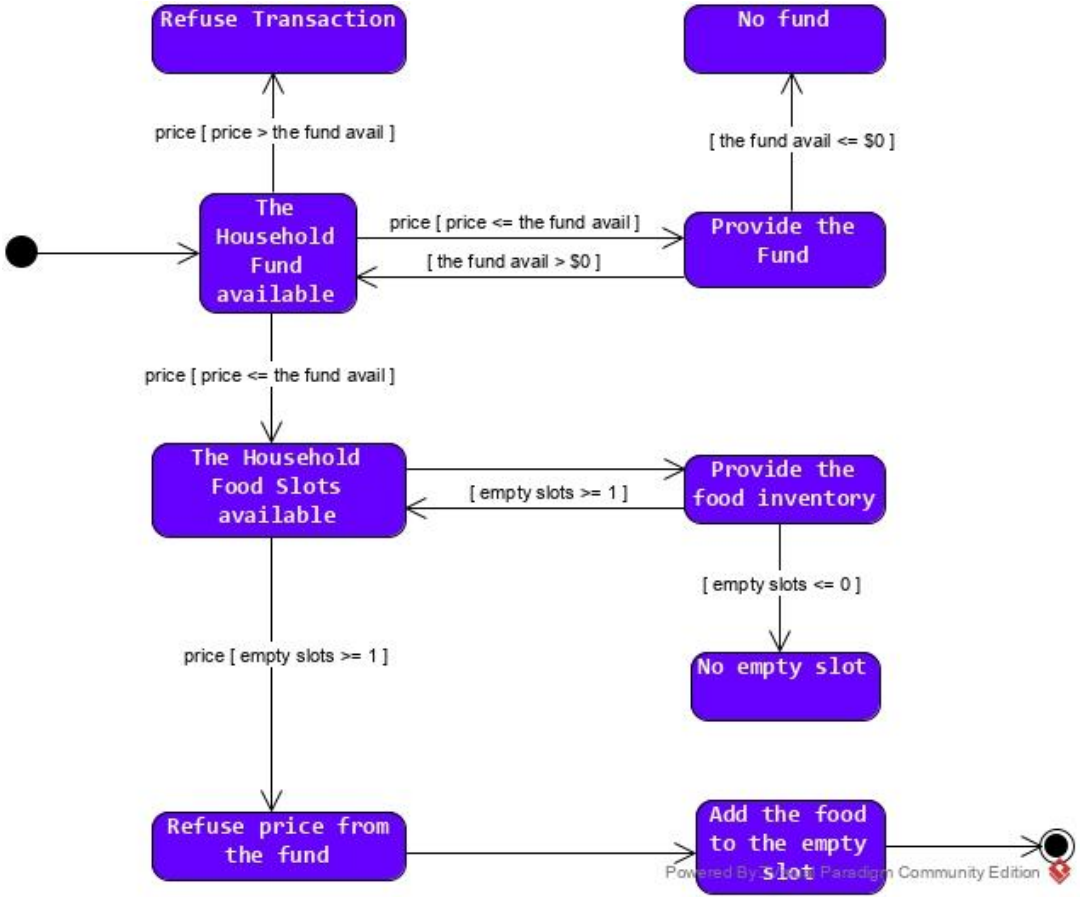
3. Eat at the Home



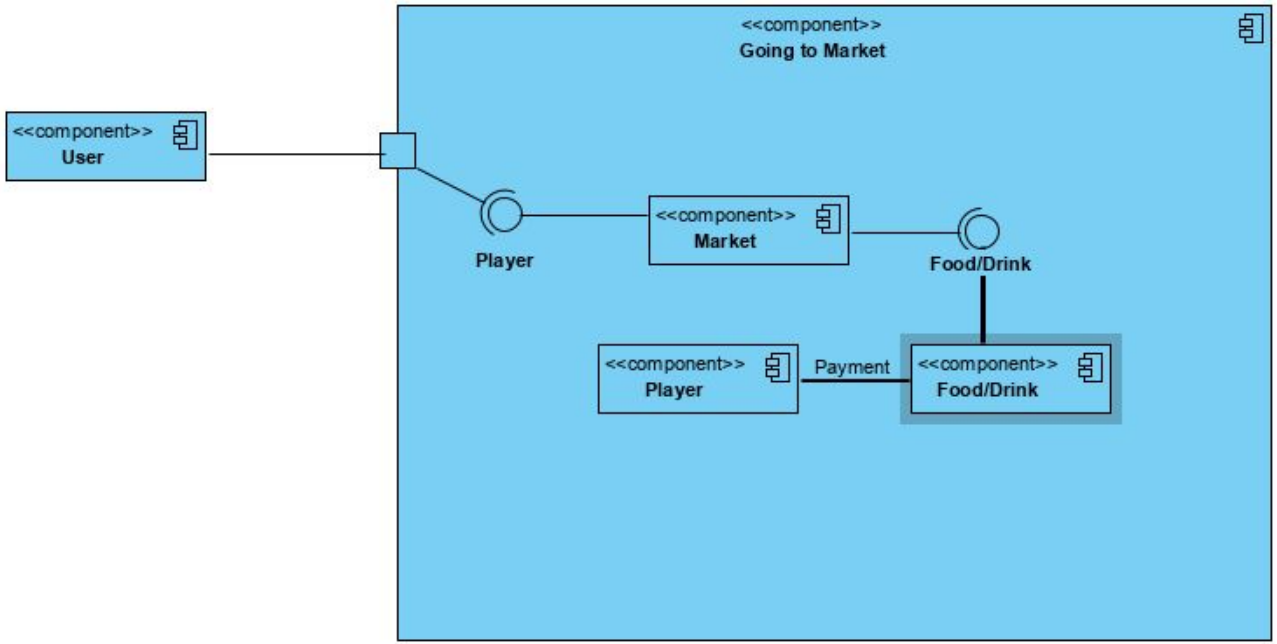
4. Go to the Theatre



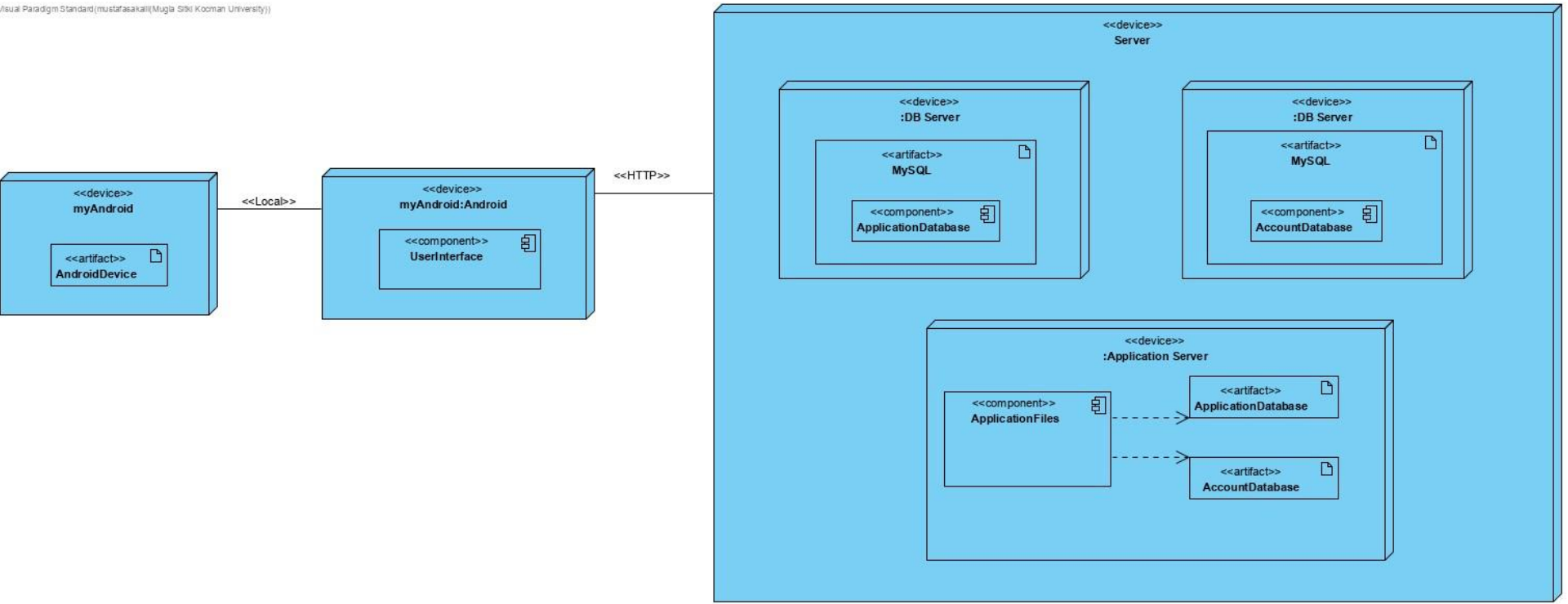
State Machine Diagram - Buy a Food



SUBSYSTEM DECOMPOSITION



HARDWARE / SOFTWARE MAPPING



OTHER DESIGN CONCERNS

1. Concurrency
2. Data Management
3. Global Resource Handling
4. Boundary Conditions

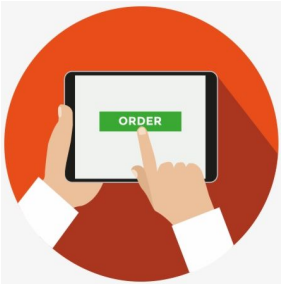
1. Concurrency

Users must first choose the player when they want to have their players do something. In other words, it is not possible for the two players to send an action request at the same time. Also, the same user is not allowed to log in to multiple devices at the same time to avoid syncing problems. Even so, we took some precautions.

1.1. Order Food with Mobilephone:

Case: When one player is in the process of getting food when another player who lives in the same house wants to buy food.

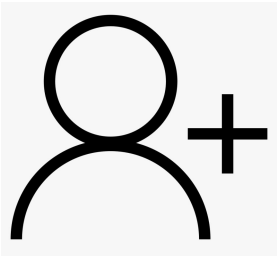
Method: In this case, the system queues the players according to the start times of the action and does not allow to start to the next player of the queue buy food before the previous is over.



1.2. Go to the Picnic:

Case: When one player is in the process of inviting other players when another player wants to do same thing.

Method: When one of the players approves to invite, the list of invitations is refreshed. When other players want to confirm the invitation, the person they choose is compared with the renewed list, and if they want to select the selected one, an error message is returned and the renewed list is displayed, and it is expected to re-select.



1.3. Eat at the Home

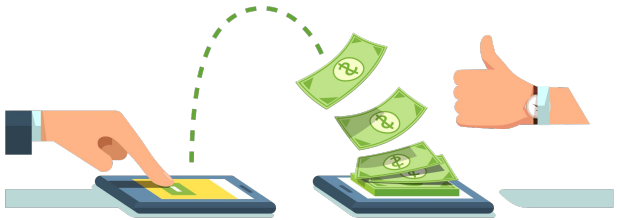
Case: Players who live in the same house choose the same food and want to confirm it to eat.

Method: As in the case while inviting other players, after confirming the choice of either of them, the list of dishes at home is renewed. If the food's stock has dropped to zero, an error message will be displayed and the list will be refreshed. However, if the stock of the food is not 0, the transaction takes place from the renewed stock and proceeded without errors.

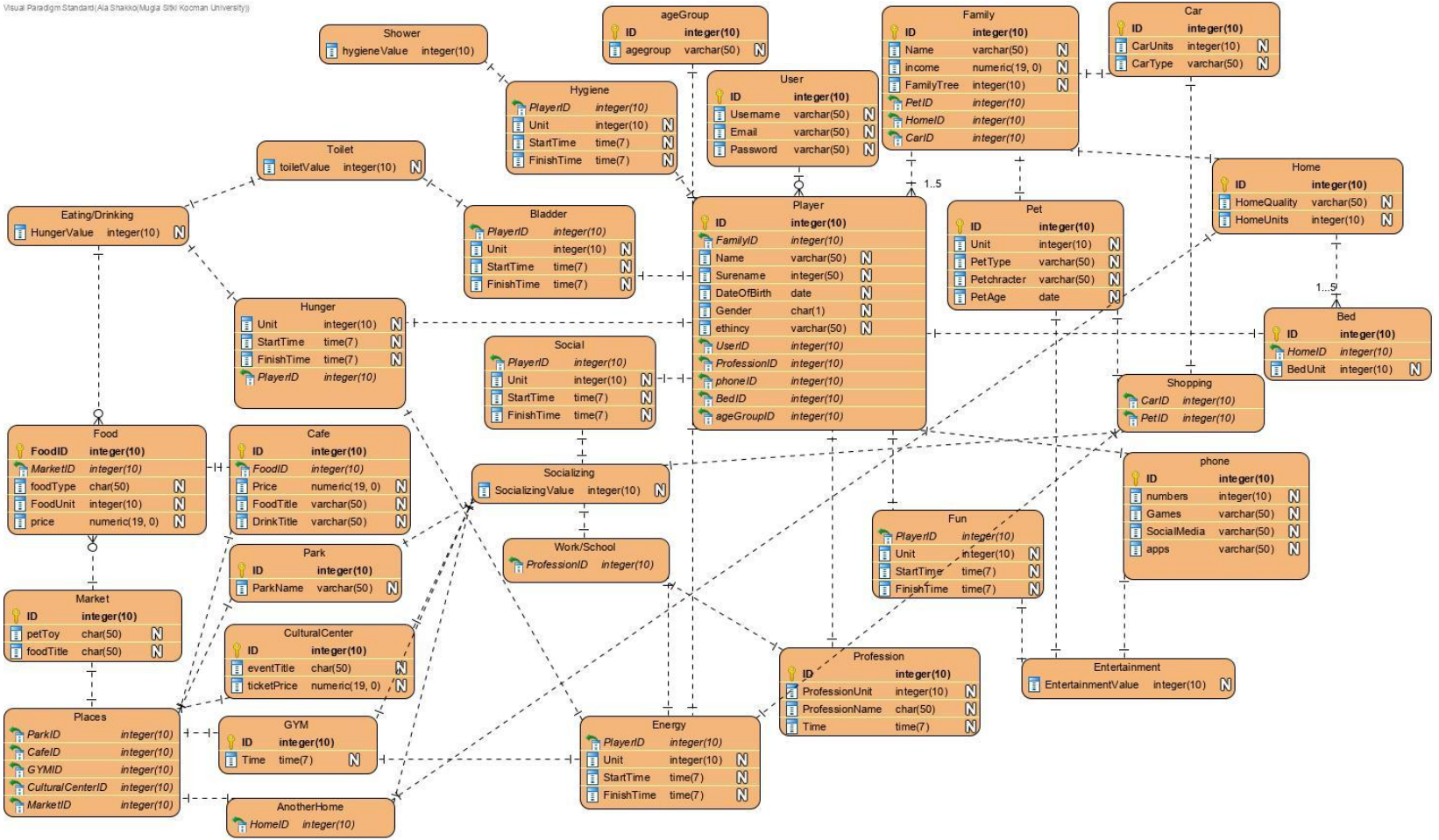
1.4. Buy a Pet

Case: Situations where two people from the same family make purchases simultaneously (consecutively).

Method: Although it is not possible for the same user to operate at the same time for two different players, sometimes there may be problems due to the connection slowness. Therefore, the system puts purchase requests in order according to their timestamps and makes transactions according to this queue.



2. Data Management



3. Global Resource Handling

Actors Classes → ↓	Home	Work/School	Bed	Pet	Gym	Shower	Hygiene	Cafe	Eating/Drinking	Car	Shopping
User	select()	select()	select()	select()	select()	select()	select()	select()	select()	select()	select()
Character	goHome() increaseEnergy()	goWork() goSchool() decreaseEnergy() increaseSocialize()	goBed() increaseEnergy() decreaseSocialize()	petAnimal()	goGym() decreaseEnergy()	goShower()	increaseHygiene()	goCafe() decreaseEnergy() increaseSocialize()	goEat() goDrink() increaseEnergy()	travel()	goShopping() decreaseEnergy()

4. Boundary Conditions

Initialization:

1) What data need to be accessed by the application to start properly ?

- Application must access application database first to load necessary files to run application

2) What data need to be accessed to load user's settings ?

- Account Database must be reached to load user's information and settings

3) What user has to do to begin play?

- User must create character to play the game

4) What the user interface show first?

-User interface first show login page, this page require username and password

4. Boundary Conditions(cont.)

Termination:

Can application run with single subsystem?

No, more than one subsystem require to work

Are subsystems notified if a single subsystem terminates?

Yes, other subsystems notifies synchronously.

How are these subsystem terminations submitted to application database?

Application database will be notified immediately after subsystem terminations.

Does the application work when it encounters an android-related issue?

No, application will be terminated and notify database about this.

4. Boundary Conditions(cont.)

How does the system behave when a node or communication link fails?

System transmits the error report and directs the application to restart

How does application behave when it face android related issue?

It requires restart and notify database about this issue. Keep user's settings saved

How does application behave when it encounters a network-related issues?

It ask for user to make sure device has a internet connection. If problem persist, application shutdown .

How does the system recover from failure?

System force application to restart

GLOSSARY

Needs: The activities player has to require in order to live. These are *energy*, *hunger*, *bladder*, *hygiene*, *sociality* and *fun*

Android: Platform where the application will run.

Concurrency: is the ability of different parts or units of a program to be executed out-of-order or in partial order.

Mustafa Sakallı

Beyza Kurt

Dilek Koçak

Ala Şakko

Presentation Video Youtube Link: https://youtu.be/frRPhH45bt0