Extra Credit: Team Case

**Our Design:** Space Trader

***The brainstorming***

- Player
- Ship
- Planet
- Trading Trade Handler
- Trade
- Database
- Consumption
- Police Encounter
- Goods
- Illegal Cargo
- Equipment
- Ship types?
- City Items
- Tech Levels
- Trader Encounter
- GUI

- Cargo
- Pirate
- Merchant
- Combat Handler
- Events: Random
- Police
- Missions
- Trade
- Resources
- Universe
- Ship types
- Pirates
- Game
- Money
- Delivery Quest
- Credits
- Basic Shield
- Trade
- Basic Laser
- Encourdcer
- System
- Profit
### ***Our CRC Cards***

#### Tech Level
- **Cards**: Name, Description, Skill, Equipment
- **Skills**: Tech Level, Equipment

#### Planet
- **Cards**: Name, Description, Skill, Equipment
- **Skills**: Tech Level, Equipment

#### Ship
- **Cards**: Name, Description, Skill, Equipment
- **Skills**: Tech Level, Equipment

#### Player
- **Cards**: Name, Description, Skill, Equipment
- **Skills**: Tech Level, Equipment

#### Card Item
- **Cards**: Name, Description, Skill
- **Skills**: Card Item

#### Equipment (Random)
- **Cards**: Name, Description, Skill
- **Skills**: Equipment

#### Card Bay
- **Cards**: Name, Description, Skill
- **Skills**: Card Bay

---

<table>
<thead>
<tr>
<th>Tech Level</th>
<th>Ship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Ship</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Planet</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ship</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Player</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card Item</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card Bay</th>
<th>Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Card</td>
</tr>
<tr>
<td>Description</td>
<td>Card</td>
</tr>
</tbody>
</table>
### Surprise Encounter

| Supernovel Encounter | Player Ship Tel.
|---------------------|-------------------
| Trouser, Shins, Arm  | Cloth, Clothes, Item |
| Head, Mouth, Nose    | Head, Mouth, Nose  |
| Body, Arms, Legs     | Body, Arms, Legs   |

<table>
<thead>
<tr>
<th>Sneak: Smell, Taste, Feel</th>
<th>Player, Ship, Item</th>
</tr>
</thead>
</table>

### Player Encounter

<table>
<thead>
<tr>
<th>Attack Player</th>
<th>Player Ship, Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Player</td>
<td>Player Ship, Room</td>
</tr>
<tr>
<td>Attack, Sneak</td>
<td>Player Ship, Room</td>
</tr>
</tbody>
</table>

### Crew

<table>
<thead>
<tr>
<th>Display The View On</th>
<th>Crew</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree House</td>
<td>Planets</td>
</tr>
</tbody>
</table>

### Trade Market

<table>
<thead>
<tr>
<th>Item, Ship, Location</th>
<th>Item, Ship, Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyable, Item, Item</td>
<td>Buyable, Item, Item</td>
</tr>
</tbody>
</table>

### Universe

<table>
<thead>
<tr>
<th>Holos, Planet</th>
<th>Planet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generators, Planet</td>
<td>Planet</td>
</tr>
</tbody>
</table>

---

- The content appears to be a table or list summarizing various elements related to a game or role-playing scenario, including encounters, player actions, and market items.
- The sections labeled 'Supernovel Encounter' and 'Player Encounter' suggest different types of interactions with entities or elements in the game.
- The 'Trade Market' section lists items available for trade, indicating a market or shopping aspect of the game.
- The 'Universe' section includes entries for holos and planets, suggesting a setting with a variety of locations or entities.

---

- The document seems to be a planning aid or reference for a role-playing or strategy game, detailing different scenarios and interactions.
- The handwriting indicates a personalized or detailed approach to the game, possibly for a specific scenario or map.
**Scenario 1:**

Bob starts Space Trader. He selects to start a new game. He distributes skills and chooses as a name TheBobOne. Starting from planet earth, Bob buys 20 units of waters and fills the ship with fuel. He travels to planet Mars, which has high police activity. Along the way he was pulled over by Officer Sally who wants to check his cargo. Bob
decides to submit to the cargo check. After the cargo check was clear, Bob continues to planet Mars. Bob sells all his cargo items in Mars for a profit 1019 credits. With the new credits he decides to buy a new cargo bay. By him buying the new cargo bay, he now has added five cargo item slots to his ship. Bob decides to travel to planet Saturn. On his way to Saturn, he encounters a trader who wants to trade 500 of RUBBER for 400 credits. Bob decides to buy just the half of the RUBBER. Bob is stopped again by a pirate. Knowing his ship can stand a chance against the pirate ship. He surrenders his ship to the pirate. So, the pirate takes away all the items in the cargo bay and lets Bob go. After arriving at Saturn, Bob decides to save and quit the game.

Scenario 2:

Moh starts a new game. He chooses to name himself Moh. For skills, he chooses 3 pilot points, 4 fighter skill points, 9 trader points and 4 engineer skill points. By default, Moh had a full tank enabling him to fly 14 parsecs, strength of 100% and an empty cargo. Currently Moh is on planet Mars. He decides to check planet Earth as he wants to buy some food. He then decides to stop by and buy some. Moh then travels to Andromeda and trade his food for water. At Andromeda, Moh decides to buy the new fast and sleek Ship S2000. Moh suddenly realizes it was 2 am and reluctantly saves and quits this awesome game.

*** Our Sequence Diagram***

*** Our UML diagram***
**Lessons learned**

1: **Attend lectures:**
Be on time (beginning of class is the most important part in Bob’s lecture)
Ask questions.
Take notes.
Read the book (If you want to Ace the test).
Run the code Bob demonstrates in class.
Treat Bob’s advises very seriously.
2: How to choose good team members:

- Please trust Professor Bob Waters, when he talks about it in the beginning of the semester (He really gives a good advice, and we learned that the hard way.)
- Don’t choose somebody just because he or she is sitting next to you in class.
- Don’t be fooled by people who talk a lot (Especially, the show offs).

Look at this person as he identifies himself:
“If I'm not coding or singing, I like to hang out, make messes, throw paper airplanes, and climb rocks.”
This person actually *didn’t code a single line in our project*.

3: How you make your Group productive:

- I suggest announcing one member to be the manager of the team (everybody has to accept his decisions and follow them.)
- Don’t be easy on each other (don’t accepts excuses especially if they seem lame.)
- Don’t have only guys in your team ***Please have some female members*** in your teams. That it would spice up the other members and motivate them work harder. Girls are hard working in general where boys are slacker in general (they always need some motivation).

4: Group meetings:

Every meeting is very important
Seriously punish who ever misses meetings
Plan your meeting before you start
Get something done before leaving
Set a time frame for the meeting in advance
Set meetings on regular basis
Set meeting time for next meeting

5: 1st meeting:

Has to be very early in the semester
Has to be at least one hour long (no less)
Obtain all contact info (names, e-mails, cell #, home#...)
Know your teammates (classes’ loads, schedules, personality, nationality…)
Set the second meeting time before you leave.

6: Coding assignments:

Use your TA and Bob advice about your design; it makes a huge difference to have a good design
Plan before you start
Assign each member a task and help each other

*Code together individually*

Test every single new method you add
Comment Code together
Code on regular basis
Consider what is easy to implement
Follow the requirements.
Break down the requirements into little pieces.

We suggest a 1st meeting form that each teammate has to fill up. Everyone is to add anything that is relevant and may be of any use to your team.

1st meeting Form:

<table>
<thead>
<tr>
<th>Name</th>
<th>E-mail</th>
<th>E-mail</th>
<th>Cell #</th>
<th>Home #</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classes taken</th>
<th>Class</th>
<th>Class</th>
<th>Class</th>
<th>Class</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How far do U live from COC(time)</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Hobbies</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Programming languages</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Best meeting times</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Expected grade</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Talk about yourself</th>
<th></th>
</tr>
</thead>
</table>

M5:
M5 is very important because it affects the outcome of your product. It is very important than M6 (the coding and testing) and the rest. The team has to discuss the way they want to implement their design, there is no right or wrong but there are easy ways and hard ways. Consider what makes scene, easy to implement, follow the requirements, and look ahead. What limits the user has, what he can do, what cannot do, what can damage your application.
You need much more scenarios than the required number to test your design and see if it solves most cases. Play your CRC cards carefully and modify them as you play. A good thing to do is to get Bob and TA’s opinions, it is very profitable. User Interface Screen Prototypes, Contract, and Exception Handling were not worth so many points but they are as important as the other parts of M5, so treat them as such.

**Things that helped us & things that hurt us.**

**M4 & M5:**

1:  
Our team was ahead of all other teams, we started very early and we got so much done way before the deadlines.

We asked TA’s and Bob’s for their opinions. They were very helpful. 
We did more scenarios that required and we played CRC cards a lot, and that helped us identify areas that we did not consider or overlooked during brainstorming. 
We did most of M4 and M5 together, we met before every class and we made sure to show our results to our TA.

2:  
We used some ideas that most of our team did not agree on. (1 against 3)
We got way too confident after M5.

**M6 & M7:**

1:  
We started early  
We used tools in Visual works such as high lighting code …  
We forced commenting our code

2:  
We had one group member missing meetings.  
We did not plan before starting to code. (We did not talk about methods we needed, methods would have made our lives easier)  
We assumed that we are on the same page  
We started coding individually.  
We divided the work all wrong (based on requirements)

**M8:**

1:  
We considered most functionaries and showed them in our GUI  
We used as a guide line previous semesters User interface evaluations  
We made use of the other team comments  
We thought of the evaluators as our products testers
2: We did not talk about simple things like colors, buttons, positions… We did not consider the audience or the product target. We did not keep in mind the simplicity of GUI.

**M9:**
1: We took notes in class, they were very useful
The TA was very useful to us

2: We did not start early
We did not plan ahead what we want to do, we just started coding. It took us more than what we expected
We coded this part individually

***Our Initial User Interface***