CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

LAB 7

Objectives

1. Identify the entities of a detailed design for your project.
2. Create detailed designs for a subset of the entities of your project.
3. Assess your team's technical capability compared to the technical needs of the project.

Designers need to specify the details of the entities that make up the system. These definitions should be sufficiently detailed that the design can be given to a developer and the developer can create the entity as envisioned by the designer.

Once your team starts to develop a design, you should also be developing a better understanding of the technologies and skill levels needed to build the product. As a separate task, this lab will also provide a chance for you to assess your team's capability to work on the project and identify learning or skill development you may need.

Procedure

Step 1 – Draft a list of entities for your project

You should consider the following types of entities:

- Screens (or Web pages)
- Database tables
- Files (e.g., data that is stored as part of the system but not stored in a database)
- Code (modules, objects, or functions)

Use Figure 7-1 to list all the system entities that you can identify. A good way to start is to pick one area and focus on that. For example, if your system has a significant user interface, start by trying to name all the screens that would comprise your interface. For each entity you list:

- Enter a type, e.g., "screen"
- Give it a meaningful name, e.g., "CustomerProfile"
- Provide any short notes or explanation needed to identify the screen, e.g., "This screen captures customer information and preferences."

Step 2 – Create detailed designs for at least 4 of your entities.

You will not be able to design all the entities of your system in this lab, but this step will get you started. Pick 4 entities that you think you understand the best at this point, and create a design for them. Every entity should have a name, type, and design details. Templates are provided to help you create detailed design for screens, database tables, and code functions.

Step 3 – Review your detailed designs.

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

After creating your designs, review them for completeness and clarity. Ask yourself this question: "If I was the developer and a designer handed me this design, would I know what to build without needing to ask a lot of questions?

If you have created the design entities as a team, set them aside for a few minutes before review each one. If you have worked in sub-groups within your team to create the designs, then exchange designs so the reviewer is a different person than the creator of a design.

Revise your designs based on the review.
Step 3 – Assess your team's capability to complete this project.
Once you have an architectural overview and the beginning of a design, you should be able to assess capability and identify things that someone on the team may need to learn. Use Figure 7-5 to summarize this information.

3.A – List the technologies you need for your project using the column on the left. Consider things such as programming languages, operating systems, specialized data sources, software libraries, support tools, and hardware.
3.B – List each team member at the top of a column, and then evaluate that person's knowledge of the technology in each row. For the column for each team member, use the following values:

1 – No knowledge or not much relative to the needs of this project
2 – Enough knowledge to accomplish part but not all of this project
3 – Knowledge probably sufficient for this project

3.C – Discuss within your team how you will start to gain capabilities that you are missing. You do not need to turn in results of this discussion in this lab, but will need to address this in the coming weeks.

What to Turn In

In order to obtain full credit for this lab, ***each team*** must turn in:

1. Figure 7-1 – Possible System Entities - Sagar Patel
2. Detailed designs for at least 4 entities in your system. Use the templates in Figures 7-2 through 7-4 to get started.

Figure 7-5 – Team Capability Assessment

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

3.

Figure 7-1 – Possible System Entities

Product: Cards Against Telegram
Team: 03
Date: 11/ 04/ 2016

Type	Name	Description or Notes
Screen	Game within Telegram Group's Chat	The game screen would be integrated within the respective Telegram chat group. The group's chat is where the game will take place, hence not needing any other screen.
Database	Player Stats	Database used to hold the high scores for each player across all Telegram games to be used for the public high score leaderboard
Files	Cards	Gson will be used to serialize Java card objects directly into JSON objects
Code	getHighScore()	Function will be ran to pull the player high scores across all rounds/games within a player's Telegram groupchat
Code	getRoundRank()	Ran at the end of every round within a game to determine the order of player round ranks chosen by the Czar
Code	getBestPlayers()	Ran to pull the high scores across all Telegram groupchat games and return the highest player scores

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

Figure 7-2 – Template for Detailed Design for a Screen

Name: Game Screen

Type: Screen

Purpose: This screen is needed to meet requirement <fill in the requirement number here>

Description: The game screen would be integrated within the respective Telegram chat group. The group's chat is where the game will take place, hence not needing any other screen.

The screen contains the following elements:
The Question for all the players.
The responses of other players.
Your response.
The responses you still have left.

Layout:

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

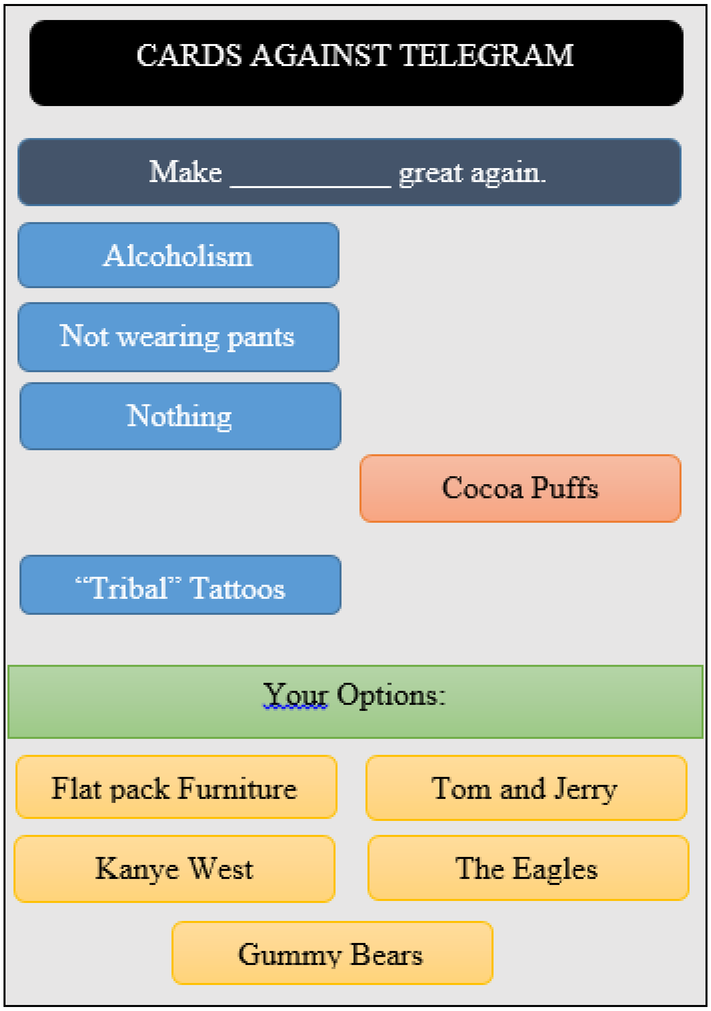

Figure 1 - Game Screen

Figure 7-3 – Template for Detailed Design for a Database Table

Name: players_data

Type: Database Table

Purpose: This table is needed to meet requirement Database Design.

Description: This table will represent the stats of the players playing the game.

Table Contents:

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

Data Element Name	**Data Type**	**Key**	**Notes**
id	int	PK	Primary Key of the database.
telegramID	int	Index	Internal ID (Telegram ID)
games_played	int	Index	Count of the time this player has played this game.
best_joke	int	Index	Count of the time this player has got the best joke

Figure <fill in a number> - <screen name> Database Table

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

Figure 7-4 – Template for Detailed Design for a Code Function

Name: getBestPlayers(int limit)

Type: Function

Purpose: This function is needed to meet requirement player statistics.

Parameters: The following parameters are used to call this function:

Name	Data Type	Notes
limit	int	How many to get

Return Type: A sorted list of the top LIMIT players.

Processing:

```
public List<Player> getSortedPlayers(){
        String query = "SELECT * FROM players_data SORTED BY best_joke LIMIT limit";
        // do query
        // aggregate data into a list
        return list
}
```

CI103

Based on our prototype results from CI102, there does not seem to be any necessary Detailed Design requirement changes or additions pertaining to our system entities, screen designs, data elements, and code functions.

Figure 7-5 – Team Capability Assessment

	Amir	**Balaji**	**Tanfe**	**Jess**	**Sagar**
Java and Telegram	3	1	2	1	
Web Design	2	1	3	1	
Web Development	2	2	2	1	
Database Design	3	1	1	1	

** The table values represent an assessment of team member capabilities. The values are:

1 – No knowledge or not much relative to the needs of this project
2 – Enough knowledge to accomplish part but not all of this project
3 – Knowledge probably sufficient for this project