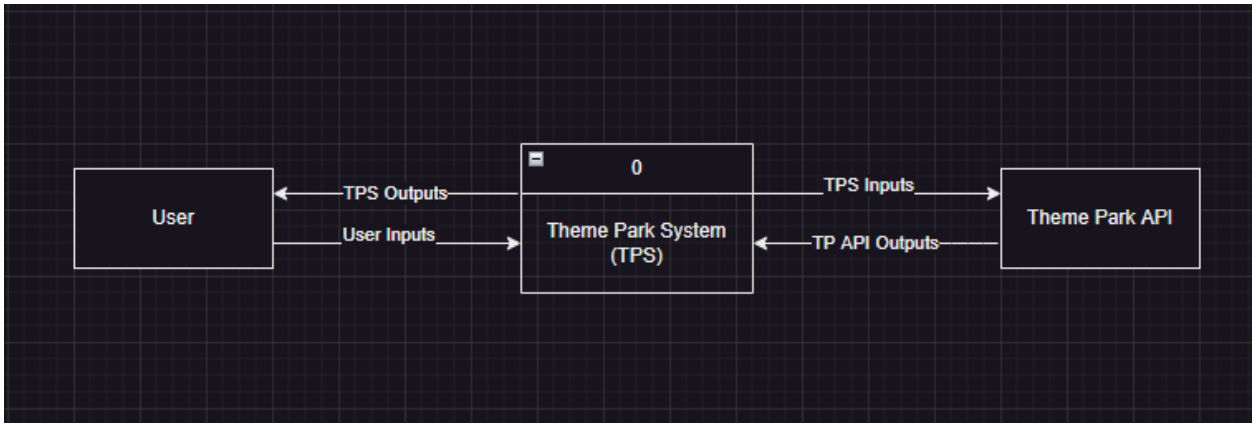
Context Diagram



Notes:

- TPS (Theme Park System) – The name of the system that runs the application and all of its functionalities
- TP API (Theme Park API) – API that a theme park (anyone that we are communicating with like Disneyland, Six Flags, etc.) has provided to the general public.
- DF (Data Flow) – Data that is in motion from one place to another.
- Process Numbers are Unique, the Process Names should be self-explanatory/explained in more detail within the description of each DFD.
- Control Flow: — <Trigger> →

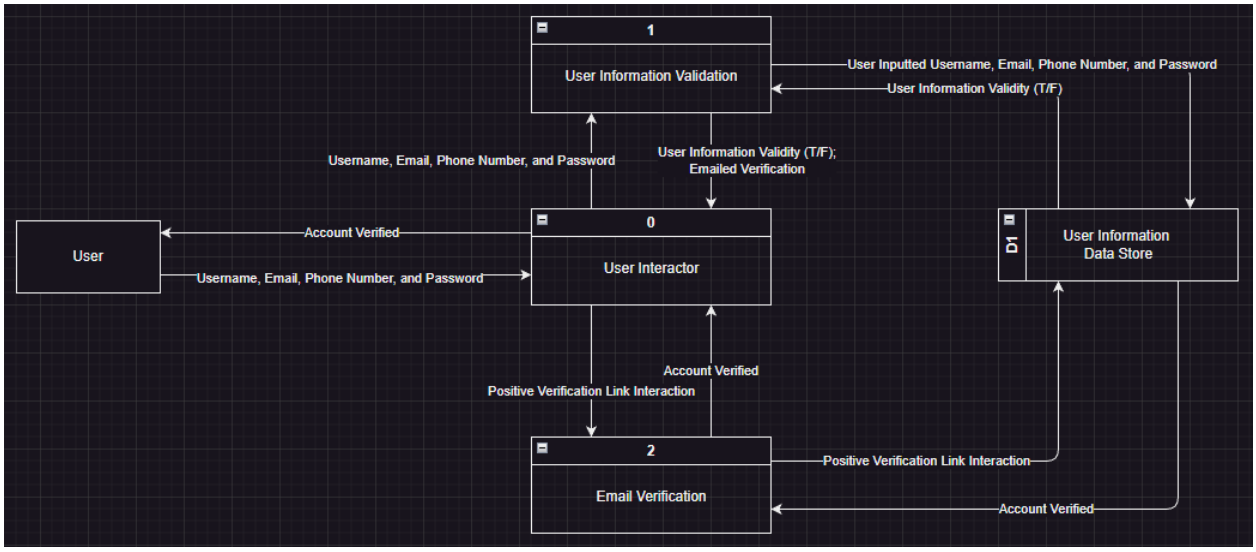
Use Case 1 – Register User

Use Case Name	Register User
Use Case ID	1
Actor	TPS User
Description	Users that want to use the application will need to have an account, which requires a username, password, email address, and phone

	number. The system will check with the database in order to see if these credentials are valid.
Trigger	User inputs user information for creating a new account
Type	External
Pre-Condition	1. TPS is online and up-to-date 2. The user is connected to the internet 3. The user has a valid email address and a valid phone number.
Normal Course	1. User information received by TPS's User Interactor 2. User Interactor sends information to User Information Validation Process. 3. TPS checks with the database to make sure that the username, email address, and phone number are unique 4. TPS verifies that the user information is valid; stores the user information 5. TPS sends the user back to the login page, stating to the user "An email has been sent to the email address that has been provided, please verify your account before proceeding." 6. The user clicks the link to verify the account, which sends a verification request to the TPS server. 7. TPS receives the verification request and verifies the user's account.
Post-Condition	1. TPS database stores that the user's account has been verified. 2. The user is logged into the TPS application.
Exception	1. The user credentials provided already exists in the database. a. Username already exists b. Email is already registered under another user

	c. The phone number is already registered under another user 2. Internet connection is lost a. User-side internet connection loss b. Server-side internet connection loss

Use Case 1 (Register User) – Level 0 Diagram



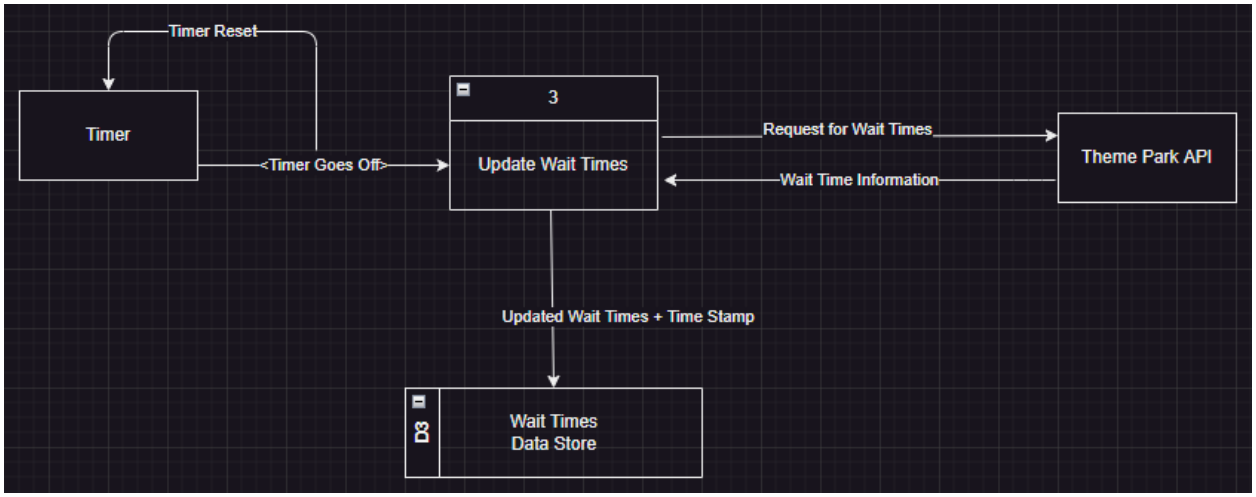
Use Case 1 Level 0 Diagram Description

The User will send their User Information to Process 0 to process where the user input should be directed towards. Process 0 determines that Process 1 is the proper destination and sends the user input to it, which will then send the User Information to D1. D1 will then send back if the User Information has been successfully stored or if it is not valid information back to Process 1. Process 1 will then send that response to the User. If the User Information is valid, then Process 1 will also send an email to the user. Once the user has interacted with the link in the email sent, a notification will be sent to Process 2, where Process 2 will tell D1 to verify the user. D1 will send that the user has been successfully verified back to Process 2, where the verification status will be sent back to Process 0, and then Process 0 will then be sent back to the user.

Use Case 2 – Requesting Wait Times

Use Case Name	Requesting Wait Times
Use Case ID	2
Actor(s)	Timer, Theme Park API
Description	The System will need to be updated frequently to provide the users with the most accurate information whenever it is requested. Therefore, the backend of the application will need to habitually update the wait times to have it quickly accessible for the front end later on if requested.
Trigger	The system requires an update (every minute) via Timer
Type	Temporal
Pre-Condition	1. The park needs to be open 2. The TP API needs to be actively open to requests 3. TPS needs to be online and up-to-date. 4. Timer is active
Normal Course	1. A request for the wait times is sent. 2. The system sends a request to the TP API 3. The TP API system will process the request. 4. The wait time information for each ride is sent back to the system 5. The database stores the information and the time it was made
Post-Condition	1. The timer starts again for one minute until the next API request is made.
Exception	1. The theme park's API is unexpectedly offline 2. Certain rides are not available for use. 3. The system is offline

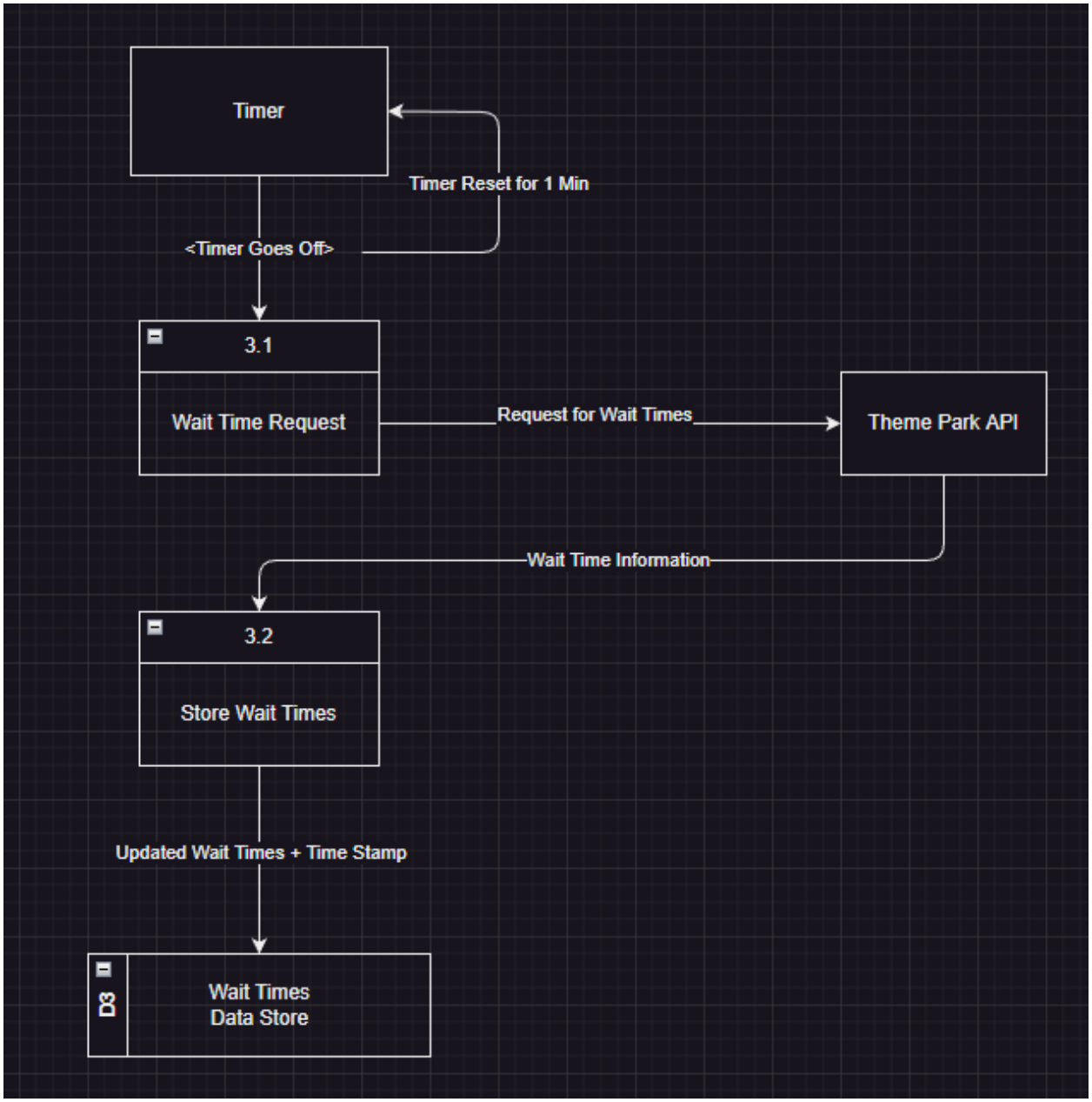
Use Case 2 (Requesting Wait Times) – Level 0 Diagram



Use Case 2 Level 0 Diagram Description

Every minute, the Timer will notify Process 3, which will prompt Process 3 to request for an update on wait times from the Theme Park API. TP API will then send the updated wait times back to Process 3, where it will send the updated information and a time stamp to D3.

Use Case 2 (Requesting Wait Times) – Level 1 Diagram Process 3



Process 3 (Update Wait Times) Description

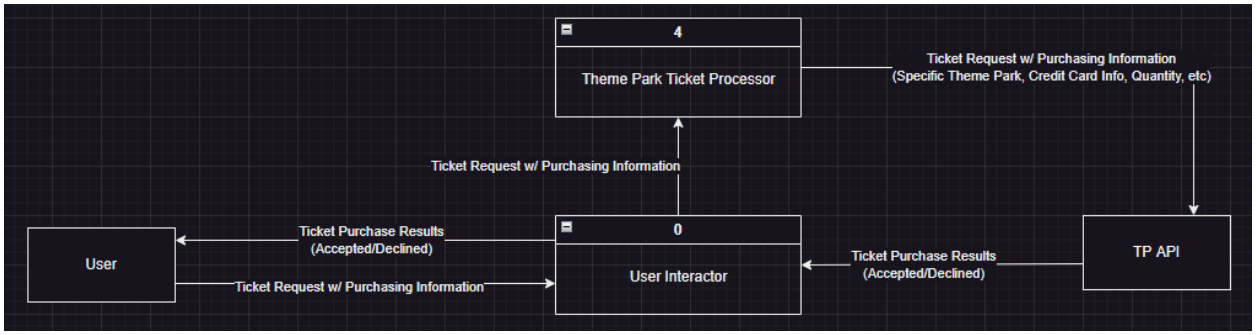
When the Timer rings, it will notify Process 3.1 which will send an API request to the TP API. The timer will set itself to ring in a minute. TP API will then send the most recent wait time information to Process 3.2. Process 3.2 will send the updated information to the D3, as well as stamping the time on the data flow of information.

Use Case 3 – Purchase Ticket(s)

Use Case Name	Purchase Ticket(s)
Use Case ID	3
Actor(s)	User, Theme Park API
Description	The User will request the purchase of tickets (quantity specified) for any specified theme park that is supported. TPS will process the information and send a request through the particular TP API, receive information on whether the request was completed or not, and then the information will be sent back to the user.
Trigger	The User sends a detailed request to purchase ticket(s)
Type	External
Pre-Condition	1. The theme park needs to be open 2. The TP API needs to be actively open to requests 3. TPS needs to be online and up-to-date. 4. Specified theme park needs to have available tickets
Normal Course	1. A purchase request will be sent from the user to TPS 2. The system sends a request to the specified TP API 3. The TP API will process the request and send back result to TPS. 4. TPS will then send result to the user with all necessary information provided by TP API
Post-Condition	1. The timer starts again for one minute until the next API request is made.

Exception	1. The theme park's API is unexpectedly offline 2. Tickets are not available for specified theme park 3. The system is offline 4. Purchasing conditions have not been met a. Insufficient Funds b. Invalid Credit Card information

Use Case 3 (Purchasing Ticket(s)) – Level 0 Diagram



Use Case 3 Level 0 Diagram Description

The User will request the purchase of tickets (quantity specified) for any specified theme park that is supported. TPS will process the information and send a request through the particular TP API (in the correct format). The TP API will then receive the information sent by Process 4, process the information, and send back the result to Process 0. Process 0 will then send the ticket purchase results (whether it was accepted or declined with the reasoning) back to the user.