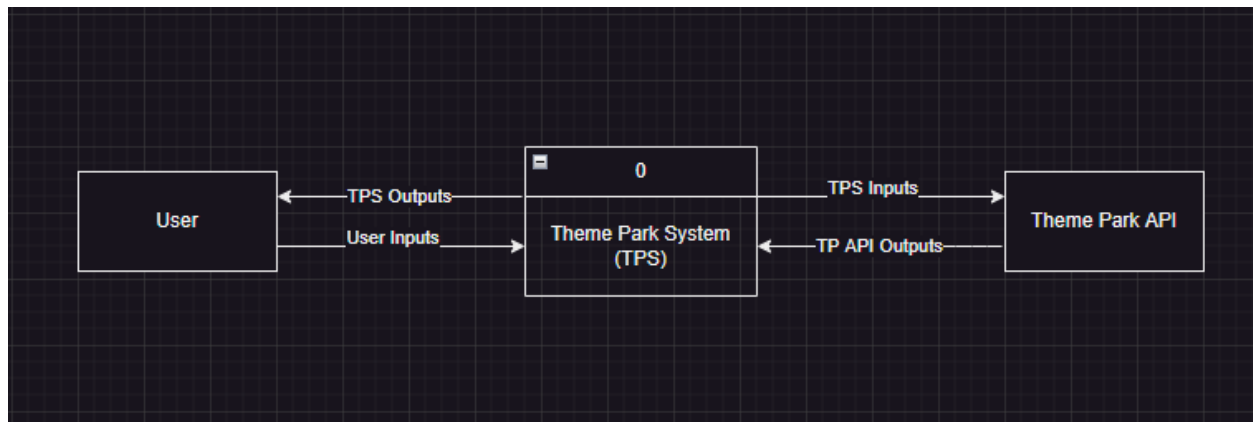


## Context Diagram



### 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.

### Use Case 1 – User Signup

**Actor:** Theme Park System 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

#### **Preconditions:**

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
2. TPS checks with the database to make sure that the username, email address, and phone number are unique
3. TPS verifies that the user information is valid; stores the user information
4. 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.”
5. The user clicks the link to verify the account, which sends a verification request to the TPS server.
6. TPS receives the verification request and verifies the user’s account.

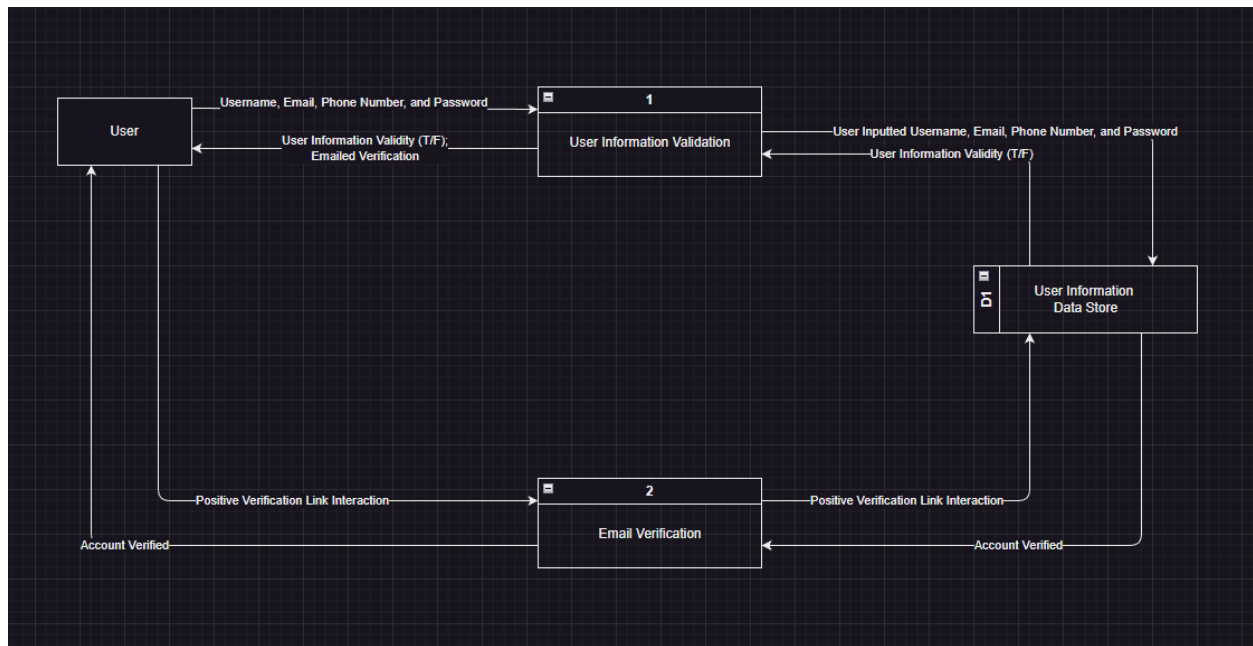
**Postconditions:**

1. TPS database stores that the user’s account has been verified.
2. The user is logged into the TPS application.

**Exceptions:**

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 (User Signup) – Level 0 Diagram



### *Use Case 1 Level 0 Diagram Description*

The User will send their User Information to Process 1, 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 then be sent back to the user.

## Use Case 2 – User Login

**Actor:** Theme Park System User

**Description:** Users that want to use the application and already have an account will be able to log into the application once they provide their user credentials (username and password). The system will check with the database in order to see if these credentials are valid.

**Trigger:** User inputs user information for authentication

**Type:** External

**Preconditions:**

1. The user already has an account with TPS (information stored in the database)

2. TPS is online and up-to-date
3. The user is connected to the internet

#### Normal Course:

1. TPS receives user credentials and checks the database for a user with the same username
2. If the username exists, it will check if the password provided is the same as the one that is stored in the database
3. Database verified user credentials, user is logged into TPS.

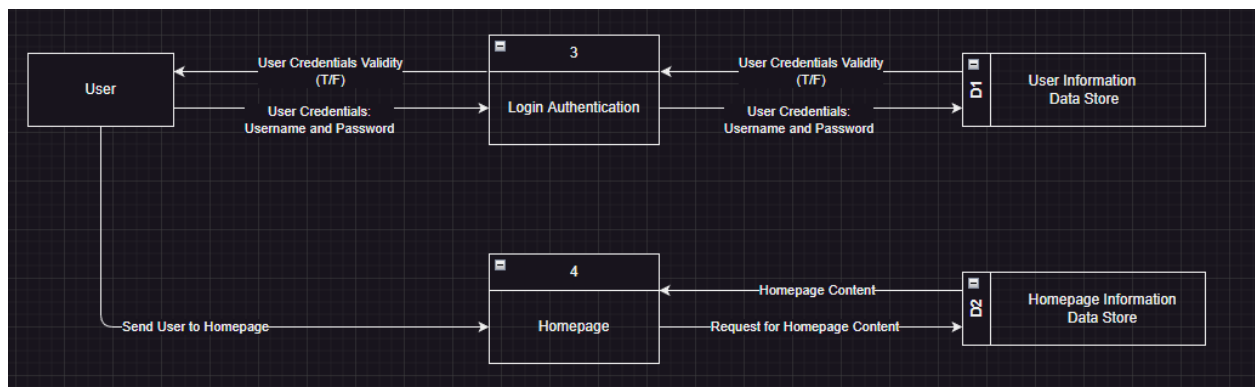
#### Postconditions:

1. Database notes when the user has logged in.
2. The user application is set to the homepage.

#### Exceptions:

2. The username provided is not present in the database.
3. The password does not match the one attached to the username provided.
4. Internet connection is lost
  - a. User-side internet connection loss
  - b. Server-side internet connection loss

#### Use Case 2 (User Login) – Level 0 Diagram



#### Use Case 2 Level 0 Diagram Description

User will send their User Credentials to Process 3, which will then send the information to D1. D1 will send back information on whether or not the information that it received was valid or not, which will be sent back to Process 3 to be sent back to the user. If the User Credentials are correct, then the user will be logged in, being sent to the Homepage on Process 4, where Process 4 will request the Homepage Contents from D2, where D2 will acknowledge the request and send it to Process 4.

### Use Case 3 – Requesting Wait Times

**Actor:** Theme Park (TP) 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)

**Type:** Temporal

**Preconditions:**

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.

**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

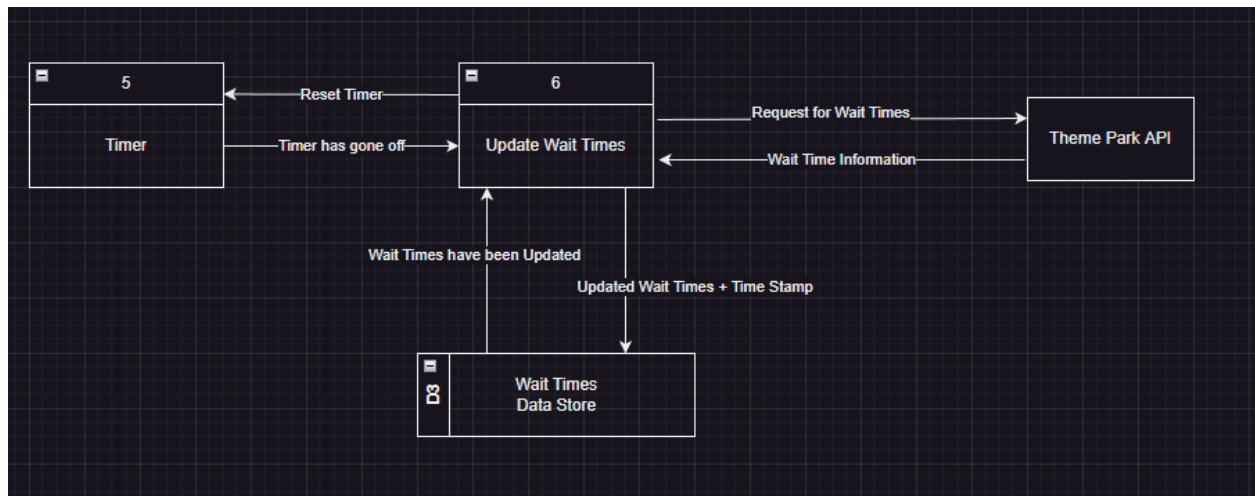
**Postconditions:**

1. The timer starts again for one minute until the next API request is made.

**Exceptions:**

1. The theme park's API is unexpectedly offline
2. Certain rides are not available for use.
3. The system is offline

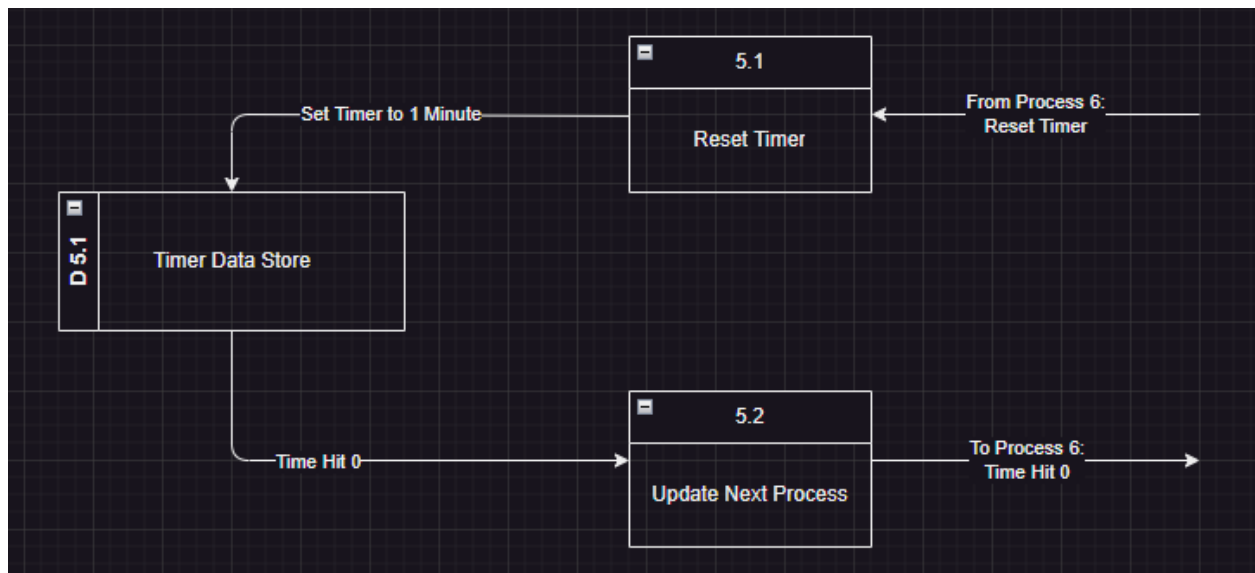
### Use Case 3 (Requesting Wait Times) – Level 0 Diagram



#### Use Case 3 Level 0 Diagram Description

Every minute, Process 5 will send a DF to Process 6, which will prompt Process 6 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 6, where it will send the updated information and a time stamp to D3. Once D3 is done storing, it will send a DF back to Process 6 so that Process 5 can start again.

### Use Case 3 (Requesting Wait Times) – Level 1 Diagram Process 5

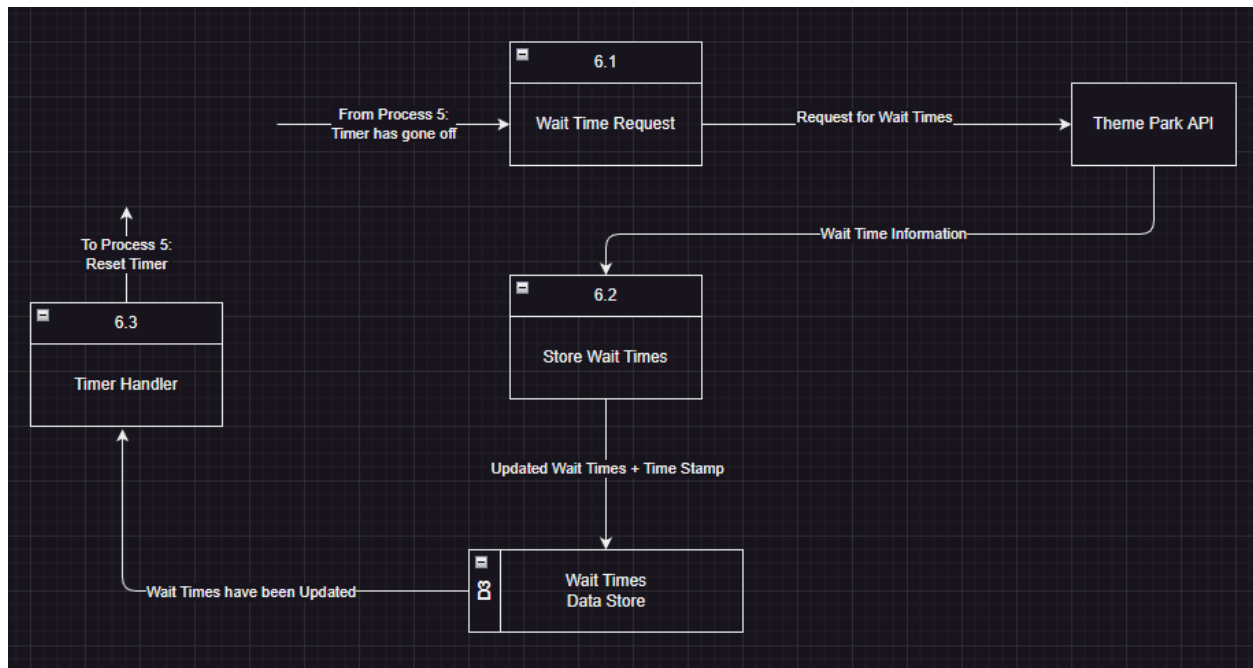


#### Process 5 (Timer) Description

When Process 6 requests a timer reset, Process 5.1 will activate, sending a request to set the timer to 1 minute. The data store will let us know when the timer goes off, sending a DF to

Process 5.2, which will tell Process 6 that the timer has gone off. This process can be updated to have more timers if needed.

### Use Case 3 (Requesting Wait Times) – Level 1 Diagram Process 6



#### *Process 6 (Update Wait Time) Description*

When Process 5 sends a DF saying a minute has passed, Process 6.1 will send an API request to the TP API, where it will then send the most recent wait time information to Process 6.2.

Process 6.2 will send the updated information to the D3. Once the data is stored, D3 will send a DF to Process 6.3 that it has completed storing the information. Process 6.3 will then send a DF to Process 5 to set the Timer to 1 minute.