Project Analysis

GiftPal – The Ultimate Gift Exchange and Reminder App

Submitted in fulfillment

Of the requirements of

CMSC 495 Current Trends and Projects in Computer Science

(Group Project)

Bazz Khurshid

Graham Primm

Kalala Kalamba

Nicholas Sink

|  |  |
| --- | --- |
| **Document ID** | GiftPal SRS -v1.1 |
| **Version Number** | 1.1 |
| **Due Date** | April 4, 2023 |
| **Classification** | Public |
| **Instructor** | Hung Dao |
| **This submission is by** | Bazz Khurshid, Graham Primm, Kalala Kalamba, Nicholas Sink |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author (s)** |
| 03/30/2023 | 1.0 | Draft Version | Nicholas Sink |
| 03/31/2023 | 1.0.1 | Added Input/Output Data and their destinations | Bazz Khurshid |
| 04/02/2023 | 1.0.2 | Added Subsystems to Functional Requirement Table, Possible Enhancements, Possible Risks and Risk Mitigation | Primm Graham |
| 04/04/2023 | 1.0.3 | Added Subsystem Diagram and Descriptions | Nicholas Sink |
| 04/04/2023 | 1.0.4 | Added conversion of input data into output data | Bazz Khurshid |
| 04/04/2023 | 1.0.5 | Specified the data processing step | Kalala Kalamba |
| 04/04/2023 | 1.0.6 | Added the context diagram | Kalala Kalamba |
| 04/04/2023 | 1.1 | Made cosmetic changes to submit | Bazz Khurshid |

**Analysis**

1. What are the input data?
   1. The following data shall be inputted as text.
      1. The user’s name.
      2. The user’s email.
      3. The name of the user’s loved ones.
      4. The user’s group name.
      5. Names of other members of the user’s group.
      6. Whether the user is an administrator.
      7. The user’s loved ones’ anniversary dates.
      8. The user’s loved ones’ birthdays.
      9. The user’s customized important days.
      10. Every individual user’s wish list of gifts.
2. What are the sources of input data?
   1. All sources of input data will be in text.
      1. These sources of input data will come from the various users.
      2. These sources of input data will be given during questions asked by the application on the terminal.
      3. There will be no input data that does not come from individual users.
3. What are the output data?
   1. Prompt questions from the application will output questions. These prompt questions will be the application seeking to gather further information from the user.
   2. Email reminders to the user.
   3. Email reports to the user on paired gift givers and gift receivers.
   4. Email gift suggestions to the user.
4. What are the destinations of output data?
   1. There will be two destinations of output data.
      1. The terminal, where the application will run. These will be the prompt questions where the user inputs all the data that will be stored in the application.
      2. The user’s email. This will be output data such as email reminders of special days, reports on the paired givers and receivers list, and gift suggestions.
5. How do we convert the input data into output data?
   1. There are two ways we convert input data into output data.
      1. The output data will be affected by the input data. The prompt menu will react differently depending on how the user answers the questions.
      2. Most of the input data will be stored in a database. When we send information to the user, such as through their email, we will grab several pieces of data from the database, depending on what information we are sending to the user, and email it to the user.

**Requirements**

|  |  |
| --- | --- |
| **Requirement #** | **Description** |
| 1 | This system shall allow users to register themselves. |
| 2 | This system shall allow users to create or register with a group by answering a prompt response once they have registered themselves as a user. |
| 3 | This system shall keep track of anniversary dates. |
| 4 | This system shall keep track of birthdays. |
| 5 | This system shall keep track of customized important days. |
| 6 | This system shall keep track of a wish list for people important to the user. |
| 7 | This system shall allow users to store their email. |
| 8 | This system shall allow users to store their name. |
| 9 | This system shall allow users to store the name of their loved ones. |
| 10 | This system shall allow users to associate special dates with specific loved ones. |
| 11 | This system shall allow users to enter their names in a gift exchange. |
| 12 | This system shall allow users to note the gift they would like to receive. |
| 13 | This system shall allow users to identify their group name (the group name will refer to the users bundled together for the gift exchange). |
| 14 | This system shall ensure that the user and all members of their group are not part of the pool of potential givers for each other. |
| 15 | The system shall randomly and automatically select a gift receiver and pair them with a gift giver once the user tells the system they will be a part of a gift exchange. |
| 16 | This system shall send an email to the users notifying them who their gift pair is and their gift pair’s wish list. |
| 17 | This system shall not duplicate user ids. |
| 18 | This system shall not send more than one email for every reminder or notification. |
| 19 | The system shall allow a user to register as an admin for the group. |
| 20 | The system shall allow the group admins to add, modify and delete users. |
| 21 | The system shall allow the group admin to set a minimum dollar amount for gifts. |
| 22 | The system shall allow the user to register others in their group. |
| 23 | The system shall present a menu to initiate the gift exchange. |
| 24 | The system shall allow the user to identify themselves when initiating the gift exchange. |
| 25 | The system shall remove the selected receiver from the list of unselected receivers and remove the giver from the list of givers once both have been paired together. |
| 26 | This system shall send a report showing paired givers and receivers to email addresses provided in user registration. |
| 27 | The system shall send a pre-configured list of gifts suggestion based on age, and minimum price with shopping links to each group once every individual gift giver and gift receiver have been paired together. |

**Specifying the Data Processing**

**Admin Registration**

* The default user registers => database is updated

**User Registration**

* User register fills out a form by providing name, email address, DOB, groups name, and additional important dates if applicable => Data is saved in the Database assigning a unique ID to each User
* The user creates a wish list for a specific event => database is updated with a wish list and user ID
* The system identifies all users belonging to the same group, based on the unique group id.

**Admin Tasks**

* Admin defines User Roles => Data is saved in Database with unique IDs for each Role defined.
* Admin defines the type of Events (Birthdays, Anniversaries…) => Data is saved in Database with unique IDs for each type of event.
* Admin adds, modifies, and deletes users from groups => database is updated (for addition, a unique ID is assigned to a newly created user).

**Track Events**

* The system tracks dates stored in the database and sends appropriate notifications based on the event type (Anniversary, Birthday…) => database is updated.

**Gift Exchange**

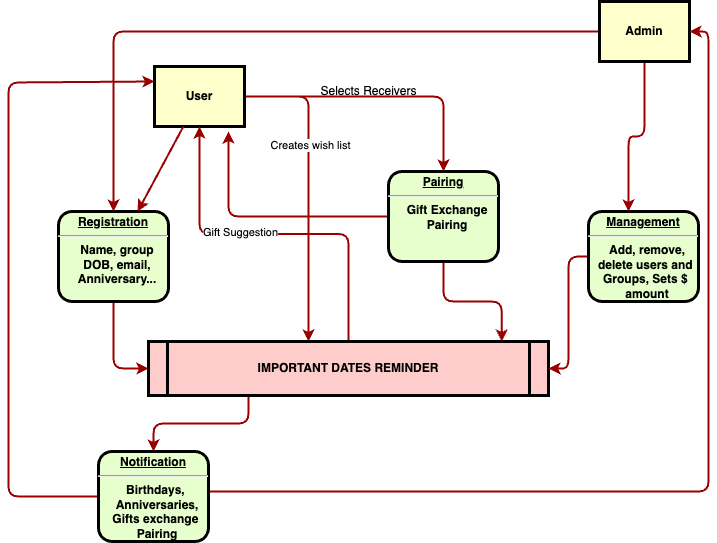
* Randomly and automatically select a gift receiver and pair them with a gift giver once the user tells the system they will be part of a gift exchange.
* Update the database with the gift exchange information.
* The System sends the saved wish list of a receiver to the giver. => Database is updated.

**Avoiding self-gifts**

* The system uses stored data to identify all users of the same group and ensures that they are not picked when one of the group members is doing the selection.

**Context Diagram**

Data between the app and the outside system:



**Subsystem Diagram**Diagram, schematic

Description automatically generated

**Descriptions of the subsystems:**

1. *User Interface Subsystem*: This subsystem is responsible for presenting an interface that enables users to register, enter important dates and names of people to be tracked by the application, add gift preferences, register users in their gift exchange group, and initiate the gift exchange. It allows users to view their gift pair and their gift preferences. The User Interface Subsystem interacts with the Administration Subsystem to register or delete users. It also interacts with the Gift Suggestion Subsystem to provide gift ideas based on the user's preferences and with the Email Notification Subsystem to notify users about their gift pair and their gift pair's wish list.
2. *Gift Suggestion Subsystem*: This subsystem is responsible for generating gift suggestions based on the user's preferences. It uses machine learning algorithms and user data to provide personalized suggestions. The Gift Suggestion Subsystem interacts with the Database Subsystem to retrieve information about the available products and their characteristics.
3. *Email Notification Subsystem*: This subsystem is responsible for sending email notifications to the users to inform them about their gift pair and their gift pair's wish list. It receives information about the user's gift pair and their wish list from the Database Subsystem and delivers the email to the application users.
4. *Administration Subsystem*: This subsystem is responsible for managing the users of the application. It also allows administrators the ability to add, modify, and delete users, as well as set a minimum dollar amount for gifts. The Administration Subsystem stores and retrieves user data and manages gift exchange groups. It communicates with the User Interface Subsystem to initiate the gift exchange and with the Email Notification Subsystem to send reports showing paired givers and receivers to email addresses provided in user registration.
5. *Database Subsystem*: This subsystem is responsible for storing and managing data related to users, gift suggestions, and other system entities. It provides a centralized and reliable storage solution for the system's data. The other subsystems interact with the Database Subsystem to retrieve and store data as needed.

**Subsystems to Functional Requirement Table**

|  |  |
| --- | --- |
| **Requirement #** | **Subsystems** |
| 1 | User Interface Subsystem, Administration Subsystem, Database Subsystem |
| 2 | User Interface Subsystem, Administration Subsystem, Database Subsystem |
| 3 | Database Subsystem |
| 4 | Database Subsystem |
| 5 | Database Subsystem |
| 6 | Database Subsystem |
| 7 | User Interface Subsystem, Database Subsystem |
| 8 | User Interface Subsystem, Database Subsystem |
| 9 | User Interface Subsystem, Database Subsystem |
| 10 | User Interface Subsystem, Database Subsystem |
| 11 | User Interface Subsystem, Gift Suggestion Subsystem |
| 12 | User Interface Subsystem, Gift Suggestion Subsystem |
| 13 | User Interface Subsystem, Gift Suggestion Subsystem |
| 14 | Administration Subsystem, Database Subsystem |
| 15 | Gift Suggestion Subsystem |
| 16 | Email Notification Subsystem, Gift Suggestion Subsystem, Database Subsystem |
| 17 | Administration Subsystem |
| 18 | Email Notification Subsystem |
| 19 | User Interface Subsystem, Administration Subsystem |
| 20 | User Interface Subsystem, Administration Subsystem |
| 21 | User Interface Subsystem, Gift Suggestion Subsystem, Database Subsystem |
| 22 | User Interface Subsystem, Gift Suggestion Subsystem, Database Subsystem |
| 23 | User Interface Subsystem, Gift Suggestion Subsystem |
| 24 | User Interface Subsystem, Gift Suggestion Subsystem |
| 25 | Gift Suggestion Subsystem, Database Subsystem |
| 26 | Email Notification Subsystem, Gift Suggestion Subsystem, Database Subsystem |
| 27 | Gift Suggestion Subsystem, Database Subsystem |

**Possible Enhancements**

1. Integrating with social media platforms to import more events and special days.
2. Providing suggestions for gifts based on past gift exchanges and wish lists.
3. Allowing users to create and share wish lists with friends and family outside of the application.
4. Adding a feature to track and suggest gift ideas for group events like weddings or baby showers.
5. Integrating with online retailers to make purchasing gifts easier for users.
6. Adding a reminder feature for important tasks related to the special day like ordering a cake or making a dinner reservation.

**Possible Risks and Risk Mitigation**

1. Data breaches due to unauthorized access to the database Mitigation: Implementing secure authentication mechanisms, encryption of sensitive data, regular security audits, and limiting access to the database only to authorized personnel.
2. Server crashes due to high traffic or other technical issues Mitigation: Implementing load balancing and failover mechanisms to ensure uninterrupted service, regularly monitoring the system for performance issues, and having a disaster recovery plan in place.
3. User dissatisfaction with gift exchanges due to mismatched preferences or inappropriate gifts Mitigation: Providing clear guidelines and suggestions for gifts, allowing users to communicate anonymously with their gift exchange partner to clarify preferences, and giving users the ability to report inappropriate behavior.
4. Misuse of personal data by the application or its employees Mitigation: Implementing strict privacy policies, only collecting and storing data necessary for the application's functionality, conducting regular training for employees on data privacy, and allowing users to easily delete their data from the system.