**Introduction**

For this project, the group will create a product that allows users to create and track their events all within the convenience of one application. The product will be an event planner application that the user can use to keep track of all the details of an event they may have coming up. Once they can create the event, they can track details like guest lists, budgets, to-do lists, calendar events, and other things that would be essential to know for an event. We want to ensure people can use our application for all their event needs.

**Project Organization**

|  |  |  |
| --- | --- | --- |
| **Position** | **Name** | **Description** |
| Design Lead | Xander Jewell | Ensures that the deliverables are organized and complete and leads the team in decisions made for the project. |
| Programmer | Makaila White | Ensures the code is written for the program and provides input for decisions made. |
| Programmer | Tanner Hadley | Ensures the code is written for the program and provides input for decisions made. |
| Organizer | Thomas Warenski | Ensures the product is organized and complete and follows along with the requirements for the project. |

**Risk Analysis**

***Risk Identification***

|  |  |
| --- | --- |
| **Risk** | **Description** |
| Project Size Increases Too Much (Occurred) | The group starts to add more details to the project, which complicates things and makes the application more difficult to complete by the due date. |
| Project Does Not Have Adequate Planning | Our group fails to complete adequate planning for the project, which leads to the expected failure of various parts and deliverables. |
| The Team Begins to Crunch Regularly | The team waits until the last minute to complete deliverables, which leads to crunch and team members becoming discouraged with the course of the project. |
| The Software Begins to Fail | The software we will use to create the application begins to fail by forgetting to save files, losing our progress, complicating processes, and other things that would affect the plan and progress of our project. |
| The Team Communication Begins to Lessen | As the project moves forward, the group begins to communicate less, which leads to unforeseen problems and conflict between the team due to uncommunicated changes. |
| The Team is Not Skilled Enough (Occurred) | Specific tasks for the project may require advanced programming techniques in languages developers are not fluent in, leading to the inability to finish. |

**Update:** In terms of these risks, some of them came up throughout our project. First, the project size began to increase a lot, but we were able to mitigate this by realizing that we only needed to meet the requirements. Second, we needed to be more skilled in some tasks. However, we were able to fix this by educating ourselves on Python, Tkinter, and SQLite3.

***Risk Analysis***

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Effect** |
| Project Size Increases Too Much | Low: The team has ensured we follow guidelines and do not become too big. | Serious: This could lead the group to envision a large product that is difficult to complete by the due date. |
| Project Does Not Have Adequate Planning | Moderate: There may be things we forget at the beginning of this project that could lead to problems in the future. | Tolerable: We can always add things we forgot as we will develop the project in an agile methodology. |
| The Team Begins to Crunch Regularly | Low: Each Group Member is dedicated to the project and will ensure that each deliverable receives enough time and effort. | Serious: This could lead to less interested team members, which means the project work will have less quality and effort put into it. |
| The Software Begins to Fail | Low: Replit, Google Docs, Github, and other software we will use are reliable and trustworthy. | Catastrophic: This could lead to an uncompleted project and a zero for the project we put so much effort into. |
| The Team Communication Begins to Lessen | Low: The group members have communicated regularly up to this point and will continue to do so throughout the project. | Serious: This could lead to conflict between the team and less interested and dedicated group members. |
| The Team is Not Skilled Enough | Moderate: All group members have used Python before, but it has been a while for some members. | Serious: Inability to complete a task because of lack of skill will set the group back significantly, though not irrecoverably. |

***Risk Prevention***

Schedule Risks [Lack of Planning, Team Crunching]: To prepare for these risks, the team has assigned one member to make a mockup of the application and created a board for assigning and tracking tasks. Additionally, the team has decided to meet weekly and assign tasks for each week within those meetings. We have made a target of getting all the work for each module completed before the weekend, and we are using an agile methodology so that if we need to change, we can easily do so. Furthermore, due to the planning and communication structures in place, if one person can not complete a portion of their work, as long as the team member gives appropriate notice, the team can work on their work.

Technical Risks [Lack of Skill, Lack of Communication]: The team has compiled resources for learning Python, Tkinter, and the details of the hosting service (Replit). We also have code technique communication channels for the team. To prevent communication breakdowns, we have a robust communication framework set up that is expanding continually, and we even have casual conversation areas for stress-free discussion if tensions ever get high,

Resource Risks [Systems Failure]: We will use tested hosting services. Replit has automatic cloud save, so our work will remain safe. Additionally, all team members have access to the files, so there is no risk of file corruption during data transfer between parties. The developers quickly resolve incidents with Replit.

Scope Risks [Project Size Increases]: We will prevent the project size from overly increasing by carefully fleshing out our initial plan. This method will allow us to know what needs to be done the entire way through. If we need to add functionality, however, we will be incrementally updating our program so that if we become too ambitious, we will still have a successful deliverable.

Financial Risk: N/A. We have no foreseeable costs; all software and hardware are free, or a team member currently owns it.

**Hardware and Software Requirements**

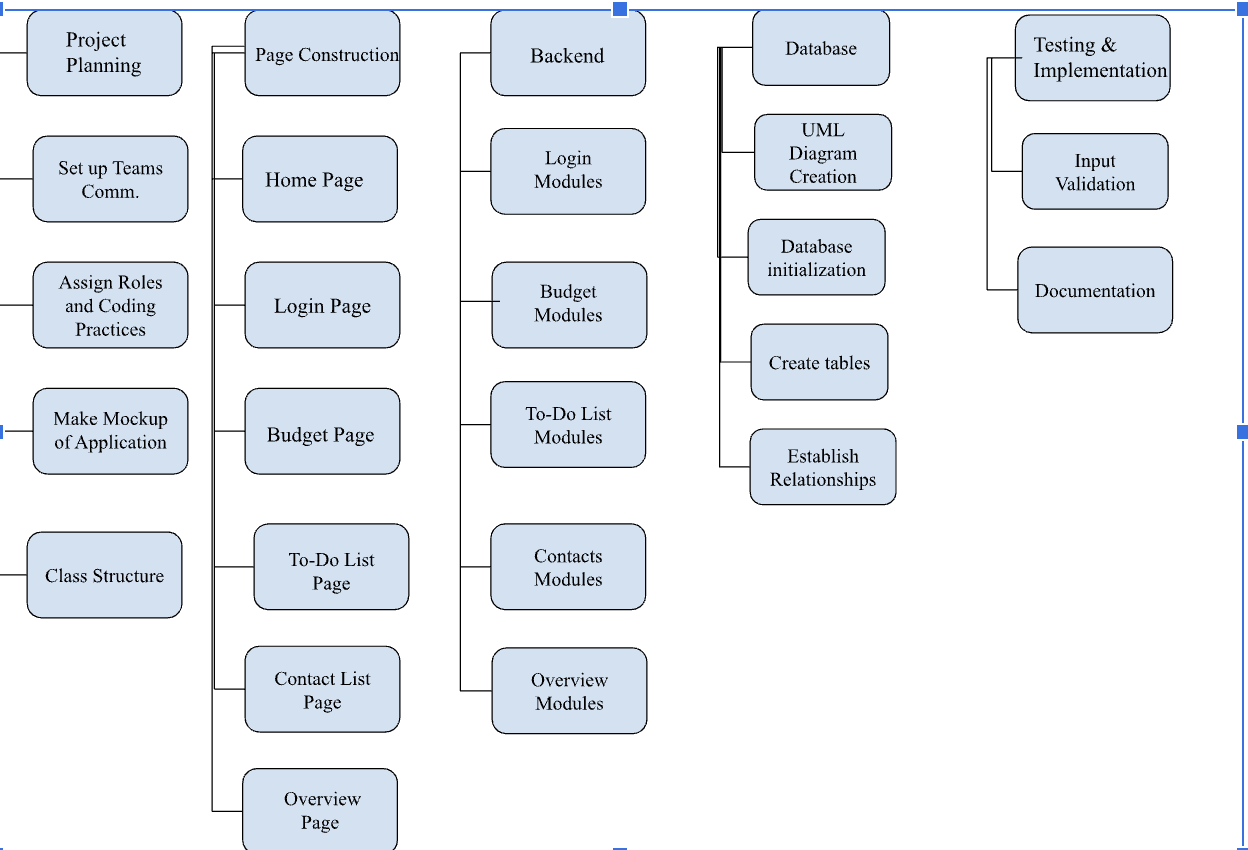
**Hardware Requirements:**

* Windows or Macintosh Computer/Laptop
* Monitor/Screen
* Keyboard and Mouse

**Software Requirements:**

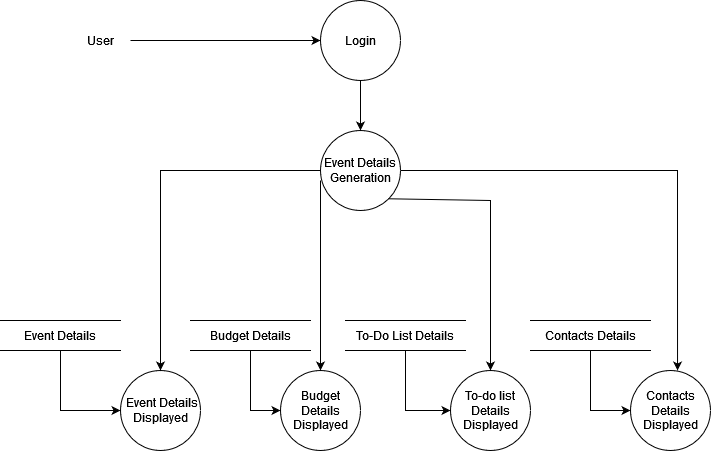
* Operating System - MacOS or Windows
* Python
* Google Docs
* Replit

**Work Breakdown**

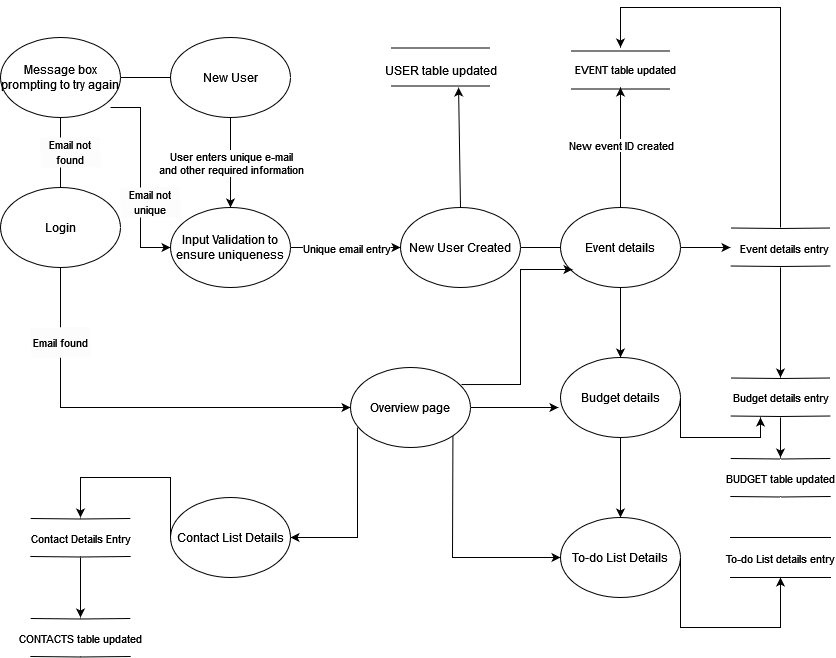


**Process Flow Diagrams**

**0-Level DFD**

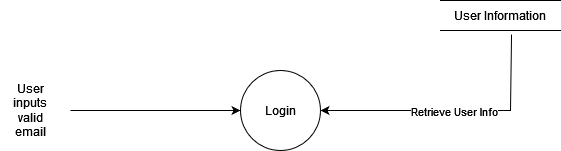
****

# **1-Level DFD**

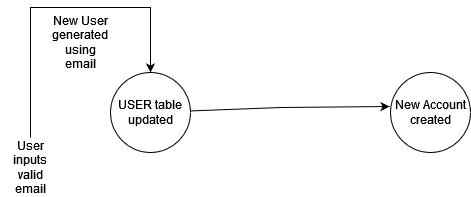
****

**2-Level DFD**

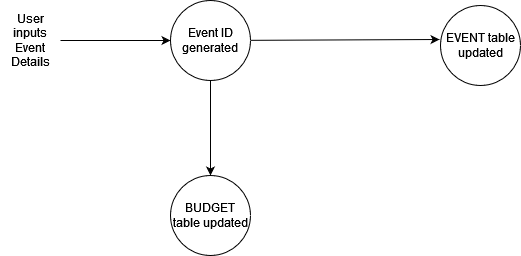
1. **Current User Login**

****

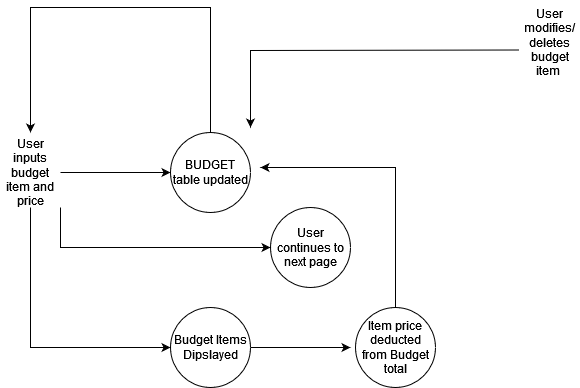
1. **New User Login**

****

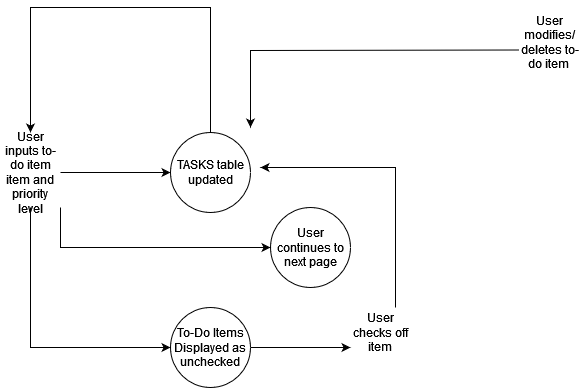
1. **Event Details Generation**

****

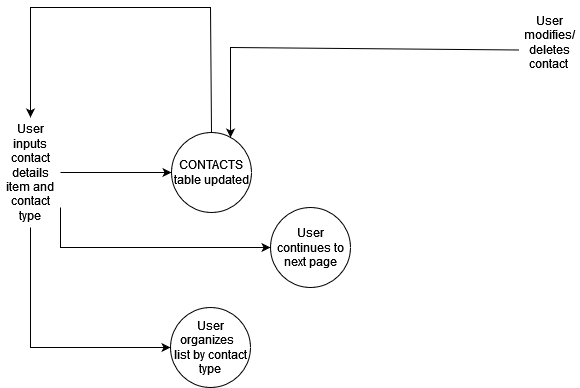
1. **Budget Details Generation**

****

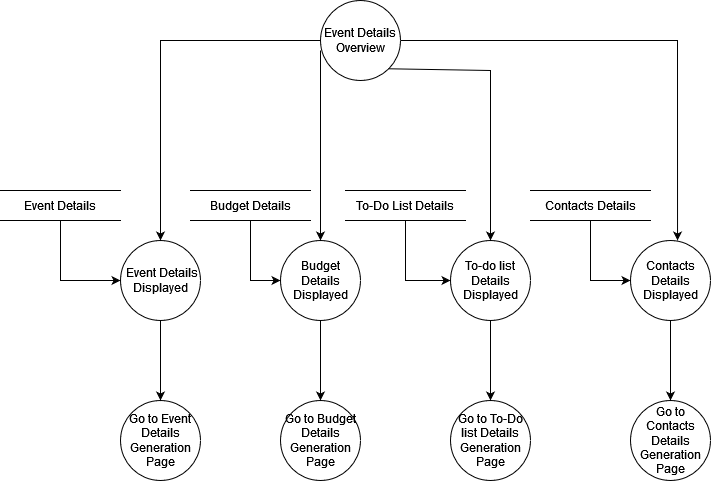
1. **To-Do List Generation**

****

1. **Contacts List Generation**

****

1. **Event Details Overview**

****

**Project Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Early Deadline** | **Late Deadline** | **Group Members Responsible** | **Completed Status** |
| Project Planning | 3/28 | 3/31 | Xander, Makaila, Tanner, Thomas | Yes |
| Team Communications | 3/21 | 3/24 | Xander, Makaila, Tanner, Thomas | Yes |
| Assign Roles | 3/28 | 3/31 | Xander, Makaila, Tanner, Thomas | Yes |
| Application Mockup | 3/28 | 3/31 | Makaila | Yes |
| Class Structure | 4/4 | 4/7 | Makaila | Yes |
| UML | 4/4 | 4/7 | Makaila | Yes |
| Home Page | 4/7 | 4/10 | Xander, Tanner | Yes |
| Login Page | 4/7 | 4/10 | Xander, Tanner | Yes |
| Budget Page | 4/14 | 4/17 | Xander, Tanner | Yes |
| To-Do Page | 4/14 | 4/17 | Xander, Tanner | Yes |
| Contact Page | 4/21 | 4/24 | Xander, Tanner | Yes |
| Overview Page | 4/21 | 4/24 | Xander, Tanner | Yes |
| Database creation | 4/5 | 4/8 | Makaila, Thomas | Yes |
| Tables | 4/12 | 4/15 | Makaila, Thomas |  |
| Relationships | 4/19 | 4/22 | Makaila, Thomas |  |
| Login Function | 4/26 | 4/29 | Makaila, Thomas |  |
| Budget Sum Function | 4/19 | 4/22 | Thomas | Yes |
| Budget Item Add Function | 4/19 | 4/22 | Thomas | Yes |
| Budget Item Edit Function | 4/19 | 4/22 | No One | Did Not Complete |
| To-Do Add Function | 4/23 | 4/26 | Makaila, Xander | Yes |
| To-Do Edit Item Function | 4/23 | 4/26 | No One | Did Not Complete |
| To-Do Mark Off Function | 4/23 | 4/26 | No One | Did Not Complete |
| Add Contact Function | 4/27 | 4/30 | Xander, Makaila |  |
| Edit Contact Function | 4/27 | 4/30 | No One | Did Not Complete |
| Response Totals Function | 4/30 | 5/3 | Xander | Yes |
| Overview Compilation Function | 4/30 | 5/3 | Xander, Makaila, Thomas | Yes |
| Logout Function | 5/1 | 5/4 | No One | Did Not Complete |
| Document Login Function | 4/28 | 5/1 | Makaila, Thomas | Yes |
| Document Budget Sum Function | 4/21 | 4/24 | Makaila, Thomas | Yes |
| Document Budget Item Add Function | 4/21 | 4/24 | Makaila, Thomas | Yes |
| Document Budget Item Edit Function | 4/21 | 4/24 | No One | Did Not Complete |
| Document To-Do Add Function | 4/25 | 4/28 | Xander, Makaila | Yes |
| Document To-Do Edit Item Function | 4/25 | 4/28 | No One | Did Not Complete |
| Document To-Do Mark Off Function | 4/25 | 4/28 | No One | Did Not Complete |
| Document Add Contact Function | 4/29 | 5/2 | Xander, Makaila | Yes |
| Doc. Edit Contact Function | 4/29 | 5/2 | No One | Did Not Complete |
| Doc. Response Totals Function | 5/2 | 5/5 | Xander, Makaila | Yes |
| Document Comp. Function | 5/2 | 5/5 | Thomas, Makaila | Yes |
| Document Logout Function | 5/3 | 5/6 | No One | Did Not Complete |
| Test Login Function | 5/1 | 5/4 | Xander, Thomas, Makaila, Tanner | Yes |
| Test Budget Sum Function | 4/24 | 4/27 | Xander, Thomas, Makaila, Tanner | v |
| Test Budget Add Function | 4/24 | 4/27 | Xander, Thomas, Makaila, Tanner | Yes |
| Test Budget Item Edit Function | 4/24 | 4/27 | No One | Did Not Complete |
| Test To-Do Add Function | 4/28 | 5/1 | Xander, Thomas, Makaila, Tanner | Yes |
| Test To-Do Edit Item Function | 4/28 | 5/1 | No One | Did Not Complete |
| Test To-Do Mark Off Function | 4/28 | 5/1 | No One | Did Not Complete |
| Test Add Contact Function | 5/2 | 5/5 | Xander, Thomas, Makaila, Tanner | Yes |
| Test Edit Contact Function | 5/2 | 5/5 | No One | Did Not Complete |
| Test Response Totals Function | 5/5 | 5/8 | Xander, Thomas, Makaila, Tanner | Yes |
| Test Compilation Function | 5/5 | 5/8 | Xander, Thomas, Makaila, Tanner | Yes |
| Test Logout Function | 5/6 | 5/9 | No One | Did Not Complete |

**Monitoring and Reporting Mechanisms**

We have used many tools for communication, collaboration, and scheduling. The group has made a Discord server to communicate and have group meetings if needed. We have also used Zoom for communication when we meet with our teacher for group project meetings with her. As for collaboration, we have used Google Docs to complete the various documents for the class. We will also use Replit so that we can all work together on the code for the application. Finally, we have been using Trello for scheduling. Trello lets us see the tasks we must complete for the week and ensures we know our progress. We will also use other software as well to help us during the duration of the process. The group will use software that will be valuable to the project and help deliver a great product.

**Appendix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Description** | **Time to Complete** | **What task relies on** | **Completed Status** | **Team Member Responsible** |
| 1. Mock-up | Creating a prototype of what the application should look like | 2-3 days | Team member communication | Yes | Makaila |
| 2. Home-Page GUI | Creating the home page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 3. Login Page GUI | Creating a login page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 4. Budget Page GUI | Creating the budget page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 5. To-do Page GUI | Creating the to-do page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 6. Contact Page GUI | Creating the Contact page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 7. Overview Page GUI | Creating the Overview page GUI for the application | 2-3 days | Mock-up, software installation | Yes | Tanner, Xander |
| 8. Database Creation | Creating the database for the application | 2-3 days | Application to store database | Yes | Tanner, Xander |
| 9. Login Function | Programming the Login into the GUI so the user can log in to an account they have made | 2-3 days | Login GUI completion | Yes | Makaila, Thomas |
| 10. Budget Function | Programming the budget page math into the GUI. This function will allow the user to see the cost of events | 2-3 days | Budget GUI completion | Yes | Makaila, Thomas, Xander |
| 11. To-do function | Programming the to-do functionality into the GUI. Allows the application to save events | 2-3 days | To-do GUI completion | Yes | Makaila, Thomas, Xander |
| 12. Contact page function | Programming the contact page functionality into the GUI. Allows the application to save favorite users | 2-3 days | Contact GUI completion | Yes | Makaila, Thomas, Xander |
| 13. Overview page function | Programming the overview page into the GUI. | 2-3 days | Overview page completion | Yes | Makaila, Thomas, Xander |
| 14. Test finished application | Test the finished application for any errors and improve the functionality of all systems | One week | Initial completion of functionality steps | Yes | Makaila, Thomas, Xander, Tanner |
| 15. Document System Functionality | Documenting the application to prove that it works and works the way we intend | Three days | Functionality of the whole application | Yes | Makaila, Thomas, Xander, Tanner |