**TEAM ORBIT**

Husain Ahmad Khan B23110006045

Muhammad Hamza Abbas B23110006096

Muhammad Hasan Khan B23110006099

Muhammad Mujtaba B23110006107

Rayyan Ibrahim B23110006137

**IDENTIFYING FUNCTIONAL POINTS:**

External Inputs (EI):

Here user can enter the data into the system

1. User Registration (email, password, details)
2. User Login (credentials)
3. Family Profile Management (adding dependents)
4. Prescription Upload (manual entry or image)
5. Reminder Creation (prescription ID, schedule)
6. Reminder Management (acknowledge, snooze, skip)

External Outputs (EO):

Here data can be produced and sent out

1. Registration confirmation
2. Authentication token + dashboard access
3. Prescription confirmation with ID
4. Prescription update/delete confirmation
5. Reminder schedule confirmation
6. Email notification/In-app alert

User Inquiries (EQ):

1. Prescription Storage (organized view)
2. Prescription Viewing
3. Prescription Search
4. Reminder Notifications retrieval

Internal File (ILF):

1. User database (accounts, family profiles)
2. Prescription database (details, images)
3. Reminder database (schedules, status, adherence logs)

External Interface (EIF):

1. Cloudinary (prescription image storage)
2. Email service provider API (for notifications)
3. Supabase (for PostgreSql)
4. Vercel (For frontend and backend deployment )

**FUNCTION, TYPE AND COMPLEXITY TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Type** | **Complexity** | **Weight** |
| User Registration | **EI** | Low | 3 |
| User Login | **EI** | Low | 3 |
| Family Profile Management | **EI** | Average | 4 |
| Prescription Upload | **EI** | High | 6 |
| Reminder Creation | **EI** | Low | 3 |
| Reminder Management | **EI** | Average | 4 |
| Registration confirmation | **EO** | Low | 4 |
| Authentication token + dashboard access | **EO** | Average | 5 |
| Prescription confirmation with ID | **EO** | Average | 5 |
| Prescription update/delete confirmation | **EO** | Low | 4 |
| Reminder schedule confirmation | **EO** | Average | 5 |
| Email notification / In-app alert | **EO** | High | 7 |
| Prescription Retrieval | **EQ** | Average | 4 |
| Prescription Viewing | **EQ** | Low | 3 |
| Prescription Search | **EQ** | Average | 4 |
| Reminder Notifications retrieval | **EQ** | Average | 4 |
| User database | **ILF** | Average | 10 |
| Reminder database | **ILF** | Average | 10 |
| Prescription database | **ILF** | Average | 10 |
| Cloudinary | **ILF** | High | 15 |
| Email service provider API | **EIF** | High | 10 |
| Supabase | **EIF** | Average | 7 |
| Vercel | **EIF** | Low | 3 |

**Unadjusted Functional Points:**

1. EI = 23
2. EO = 30
3. EQ = 15
4. ILF = 45
5. EIF = 20

**UFP = 133**

|  |  |  |
| --- | --- | --- |
| **GSC** | **Rating** | **Short Justification** |
| **Data Communications** | 3 | The system uses HTTPS, REST APIs, and email notifications |
| **Distributed Data Processing** | 4 | Uses MERN with cloud (Cloudinary) |
| **Performance** | 3 | Response time requirement |
| **Heavily Used Configuration** | 2 | Cloud deployment, multiple users, but not extreme |
| **Transaction Rate** | 3 | Frequent reminders and log updates but not enterprise scale |
| **Online Data Entry** | 4 | Users input prescriptions, reminders, profiles online |
| **End-User Efficiency** | 4 | Responsive UI, elderly-friendly design |
| **Online Update** | 4 | Users can update prescriptions, reminders, family data |
| **Complex Processing** | 3 | Reminder scheduling, adherence tracking, role-based access |
| **Reusability** | 2 | APIs can be reused for future mobile app |
| **Installation Ease** | 2 | Web-based app, easy to deploy on cloud |
| **Operational Ease** | 3 | Automated logs, reminders |
| **Multiple Sites** | 4 | Cloud-based, accessible anywhere |
| **Facilitate Change** | 4 | MERN stack allows quick updates, modular APIs |

**Sum of All GSC = 45 45**

Now calculating CAF

CAF = 0.65 + 0.01 \* Total GSC Score

CAF = 0.65 + 0.01 \* 45

**CAF = 1.05**

Adjusted FP = UFP \* CAF

Adjusted FP = 133 \* 1.05

Adjusted FP = 139.65

**Adjusted FP = 140**

**Calculating Effort:**

Effort = Size \* Productivity

Assume Productivity = 10 hours per FP

So,

Effort = 140 \* 10

**Effort = 1400 hours**

Since normally industry hours in a month are 160,

So,

1400/160 = 8.75 = 9 Person Month

Being students, it’s difficult to give 40 hours in a week so we can give roughly 9 hours in a week, that would be 36 hours/PM

Now,

1400/36 = 38.88 = 39

5 team members and 1 is developer

Industry based criteria:

1 Person \* 6 months = **6 PM**