TEMPLATE ( EMAIL TEMPLATES)

|  |  |  |
| --- | --- | --- |
| UI DESIGNING | DATA DESIGNING | CODE |
| KAUSHIK, ABHISHEK,  MAYANK, PADMAJA  UI DESIGN NEEDS TO VALIDATED BY KAUSHIK  AMISHA AND KAUSHIK -> WILL MERGE THE PROJECT STRUCTURE  **REVIEW THE PROTOTYPE (INTERNALLY)**  **SEND IT TO VIVEK TO DO REVIEW IT ONCE MORE**  MERGE THE PROTOTYPE  CREATE THE MASTER PAGE  AND THE INDEX PAGE | PRAGNYA, DEIPAYAN, ABHISH  PADMAJA  CREATE THE SCRIPTS  CREATE THE MASTER DATA  MAKE THE DATABASE  SEND IT TO ALL THE MEMBERS OF THE TEAM | AMISHA, VIVEK,VAMSHI |

|  |  |
| --- | --- |
| **MASTER DATA** |  |
| **FOR RUNNING THE PROJECT** |  |

|  |  |
| --- | --- |
| **TESTINGDATA** |  |
| **CREATING DUMMY DONORS**  **CREATING DUMMY RECEIVERS** |  |

|  |  |
| --- | --- |
| **PROJECT ARCHITECTURE** | **AMISHA, VIVEK - make it ready** |
| **DEVELOPMENT** |  |
| **INDEX PAGE & login** | **KAUSHIK (POINT OF CONTACT FOR INTEGRATION)**  **MASTER PAGE**  **(DYNAMICALLY HIDE OR UNHIDE MENUS)**  **MainMaster**  **DonorMaster**  **RequesterMaster**  **AdminMaster** |
| **DONOR MODULE** | **MainMaster**  **DonorMaster (Left side menu for donor)**  AMISHA, VIVEK  CONVERT THE HTML INTO ASPX AND DOING THE DEVELOPMENT |
| **REQUESTOR MODULE** | **MainMaster**  **Requestermaster(Left side menu for requester)**  ABHISHEK,ABHISH |
| **ADMIN MODULE & LOGIN** | MainMaster  AdminMaster**(admin side menu for requester)**  MAYANK,DEIPAYAN,PRAGYA  PADMAJA |
|  |  |
| **DESIGN , MAINTAINANCE OF DATABASE**  **MASTER DATA ,TEST DATA** | PADMAJA**(POINT OF CONTACT FOR DATABASE)** |
|  |  |

1. **COMPLETE THE PROTOTYPE**
2. DATABASE DESIGN
3. SAMPLE DATA
4. EMAIL TEMPLATES -> SENDING EMAIL USING TEMPLATES - VIVEK
5. STARTING WORK ON ONE PAGE – DONOR REGISTRATION – AMISHA

|  |  |
| --- | --- |
| PROTOTYPE TO BE SHARED WITH ALL ONCE READY | KAUSHIK – BY TOMM MORNING |
| DATABASE TO BE SHARED WITH ALL ONCE READY | PADMAJA BY SATURDAY MORNING  (ANY STRUCTURAL CHANGES NEEDS TO BE DISCUSSED BY PADMAJA) |
| INTEGRATION OF PROTOTYPE WITH PROJECT STRUCTURE  DESIGN THE MAIN MASTER  DESIGNING THE INDEX PAGE | KAUSHIK BY SATURDAY MORNING |
| START THE DEVELOPMENT BY SATURDAY MORNING | |
| BY MONDAY MORNING -> | YOU CAN SHOW SOME DEMO’S FOR THE MODULES |
| BY WEDNESDAY EVENING -> | DONOR MODULE, REQUESTER MODULE, ADMIN, MODULE |
| THURSDAY MORNING | KAUSHIK WILL INTEGRATE AND TEST IT |
| Hosting | Need to be planned |

**VAMSHI -> ONCE YOU HAVE RECEIVED THE COMPLETE PROTOTYPE. REVIEW IT AND THEN ALSO CHECK WITH SRS, AND PREPARE A REPORT**

|  |
| --- |
| COMPLETE THE DATABASE DESIGN  CREATE THE TABLES  TOMM MORNING- GENERATE THE SCRIPTS AND THE ER-DIAGRAM TO PADMAJA FOR REVIEW |

|  |  |
| --- | --- |
| STATUSID | NAME |
| 1 | REQUESTED |
| 2 | APPROVED |

**RECEIVER -> REQUEST AN ITEM**

**DONOR -> APPROVE THE REQUEST**

**ADMIN-> APPROVE**

**DISPATCHED**

**RECEIVED**

**CLOSED**

|  |
| --- |
| DLL : SHARE THE CLASSES ACROSS MULTIPLE PROJECTS |

**SERVICE ORIENTED ARCHIECTURE**

**ACCESSING THE DATA / METHODS/ CLASSES**

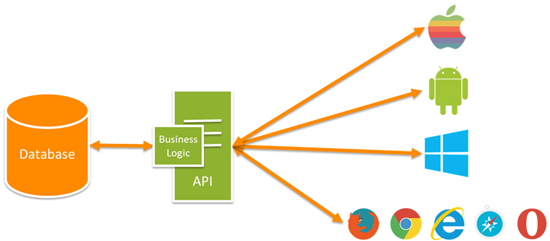
**ACROSS MULTIPLE PLATFORMS**

**SERVICE ORIENTED ARCHIECTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **WEB SERVICES** | **WCF** | **REST FULL**  **SERVICES** | **WEB API** |
| **Accessing the logic from the code behind file** | | **EXPOSE THE SERVICE IN THE WEB , WHICH WILL RETURN DATA IN JSON OR XML FORMAT WHICH CAN DIRECTLY CALL IN THE HTML PAGE WITHOUT THE CODE BEHIND FILE** | |
| **Create a proxy class of the services and use the services through the objects of the class.**  **The implementation is done in the business logic** | | **SINGLE PAGE FRAMEWORKS.** | |
| **OLD TECHNOLOGY** | |  |  |

The web api will be exposed through web and it can be accessed by different platforms

The business logic can be exposed over WEB API’S

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-56-73/2318.WithAPIArchitecture.PNG)

The web API’S HAVE TWO THINGS

|  |  |
| --- | --- |
| URL | ACTIONS VERBS |
| <http://url> | Specific action defining some specific operations  GET : select the records  POST : insert the records  DELETE : delete  PUT : update |
|  |  |

When we call the particular url with the particular action verbs, then the action is performed

In .Net framework, we have a specific templates for creating web api