

Event registration system

P4: Design Specifications



team 2

Vignesh Dhanabal

Huyen Nguyen

Quyen Nguyen

Daniel Sethuraman

1. **Executive Summary**

After completing the requirements analysis, we move to the logical design phase. In this phase, different alternative design possibilities are analyzed first. This step allows us to decide on the best design that satisfies the client’s requirements and is feasible within our project scope. We also look at the system architecture to understand the structure and components of the system we plan to build. In this design phase, the physical data flow diagram shows how the Event Registration System is actually implemented.

It is important to create user interface design and database design in the design specifications phase so that a prototype can be built accordingly. We discussed the user interface design with the client to make sure the design is user-friendly and able to provide sufficient information. In this case, the user interface design looks like a form where people can fill out the information to register for an event. The client wants the design to be really simple and includes specific mandatory fields including First name, Last name, Building and apartment number, Email, Number of participants. Meanwhile, the database design is also critical in the way that the number and content of the tables in the tables are optimized. This helps the client refer the data in the database effectively.

After all, a test plan helps check any errors in the flow of data and inconsistencies in the system. This test plan includes integration and system testing as well as test cases. After the prototype is built, system will be tested according to the test plan.

1. **Alternative design possibilities**

The first alternative is to satisfy the user requirements with minimal difference from the current system. Currently, Family Resource Center is using Google forms or paper forms to receive event registration. With minimal change from this system, the user interface design in the new system can be the same and the only thing added in the new one is the database. This means that the user interface design in both the current and the new systems contain the registrant’s information and the number of participants. The database design therefore includes two simple tables: one of registrant’s information and the other of the number of participants.

The second alternative is to provide not only the functions the client requires but also additional features the client desires. With this option, the user interface design displays more details. For example, it does not just show the number of participants but the number of adults and children along with their names in particular. Participants can also cancel the registration after they register successfully. Especially, there is a running number showing the number of slots left for events with limited slots. A reminder will be automatically sent from the system to participants when the event day is approaching. A banner can be included in the user interface design to make it more appealing and improve user experience.

The last alternative is a blend of those two above options where some advanced features of the second alternative are compromised and more features are added to the first one. Reminders and appealing banners can be excluded in order to match the project scope. The other features in the second option can be fulfilled. This is the design alternative that we choose for this project.

1. **General system logical and physical design**

To develop the system architecture for our event registration system, we focus on four main components of the system.

*System scalability and processing ability:*

The system is subject to improve in terms of scalability and processing ability. The system could process a huge numbers of registration at the same time to meet the need of all residents in University Apartments. This will help to improve the online registration speed and the efficiency of the event registration system. The system is also expected to be able to allow registration of any big events in the future that may involve the whole campus residents.

*Data storage:*

In the event registration system, we use the existing databases of Family Resource Center to store event registration information. The database in the back-end of the Family Resource Center (http://frcokstate.com/) is a new developed one with sufficient storage space for future event registration. So there is no more cost added to build or improve the database for the system. And the cost of data storage will not be the concern in the system architecture and design.

*Network traffic and connectivity:*

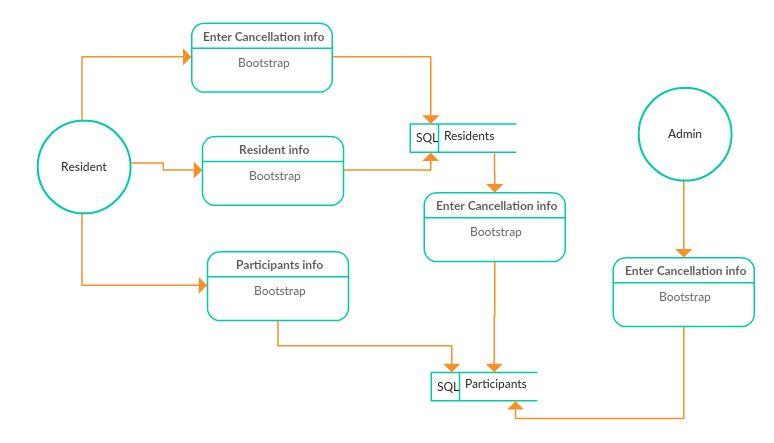
In our event registration system, network traffic is normally not very high because the maximum occupancy of an event usually small. Additionally, time required for an online registration is very short, so the probability of creating a high traffic registration is very low. However, the network traffic may increase sometimes when many residents try to register for an event at the same time. Also, the event registration activity does not require a log-in step. This may also increase the number of registrations at the same time. To take care of this situation, there is an alternative option to register for the university apartment residents which is to call to FRC office to register for the event, and the FRC staff will access the system as an admin to help residents to register.

Connectivity is not really an issue in the event registration system because the Family Resource Center uses the same network system of Oklahoma State University which is maintained frequently. All the residents in university apartments also use the high-speed internet of Oklahoma State University to connect to the registration system.

*User Experience:*

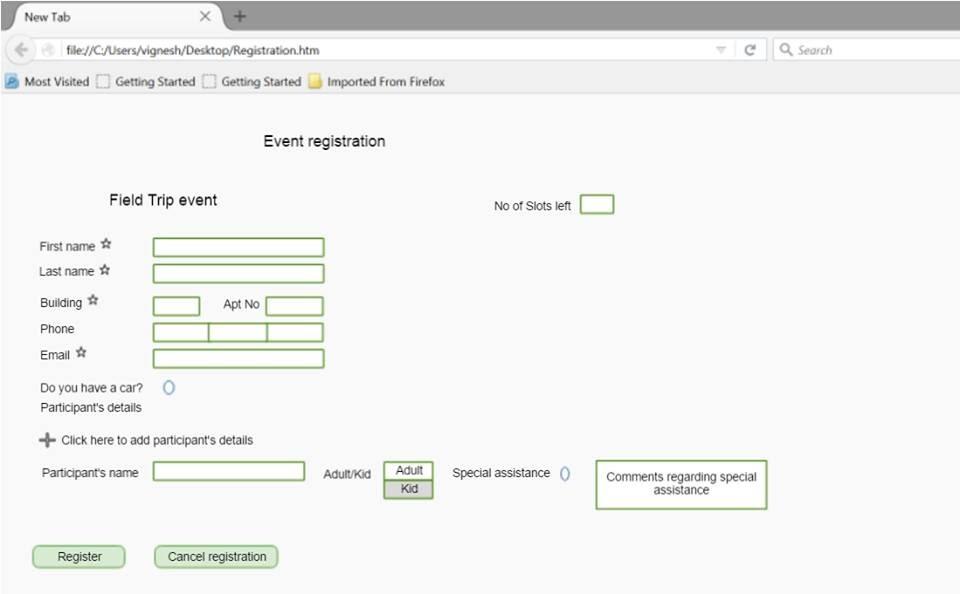
We design a user-friendly interface for event registration with which all the residents in the university apartments can view and register for an event. It is not required for the resident who registers to be an OSU student with a valid CWID. His or her spouse resident could register to their whole family to attend an event. This will help to develop the community activities in the Family Resource Center. All the tasks of verify resident will be done by FRC admin in the back-end using information of building and apartment numbers and resident information to confirm a registration.

1. **Physical data flow diagrams**

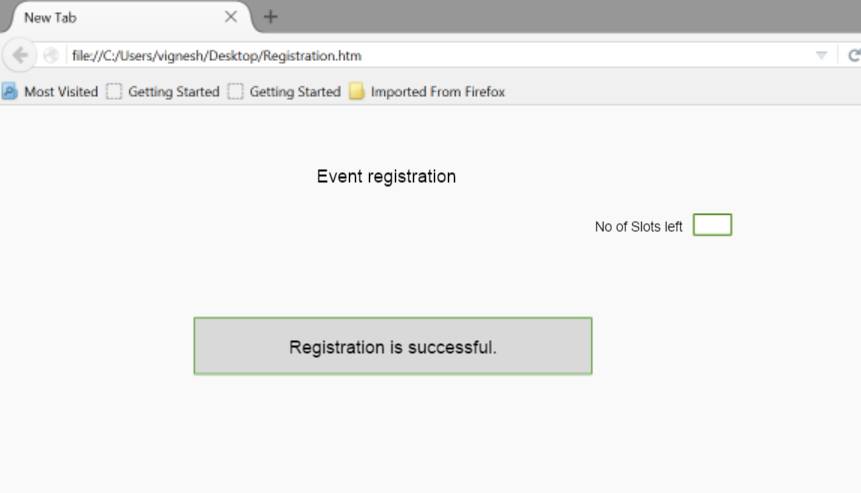


1. **User Interface Designs**

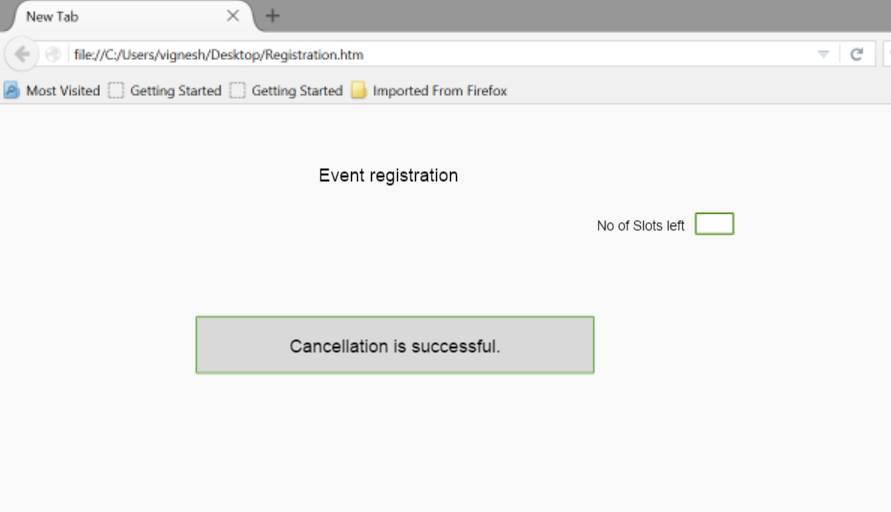
*Input*



*Output for Registration confirmation*

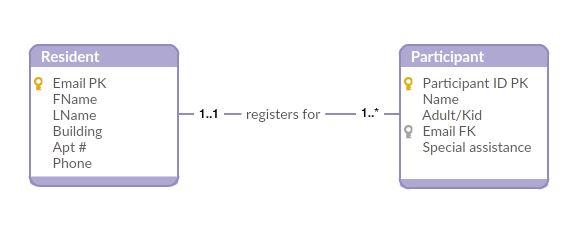


*Output for Cancellation confirmation*



1. **Database Design**

*Entity Relationship Diagram*



*Data dictionary*

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Attribute** | **Description** | **Occurrence** |
| Resident | Email | Email of registering resident | Each entry represents a distinct resident register for an existing event |
|  | FName | First name of registering resident | Each entry represents a distinct resident |
|  | LName | Last name of registering resident | Each entry represents a distinct resident |
|  | Building | Building number in university apartment | A building includes many apartment |
|  | Apt# | Apartment number | An apartment could allow two registering residents for an event |
|  | Phone | Phone number of registering resident | Each entry represents a distinct resident |
| Participant | Participant ID | System generated ID for each entry of participant | Many participants could be associated with one registering resident |
|  | Name | Name of a participant in an event | Many participants could be associated with one registering resident |
|  | Adult/Kid | Is participant adult or kid? | Each entry represents a distinct participant |
|  | Email | Email of registering resident | Many participants could be associated with one registering email |
|  | Special Assistance | Note for special assistance (if any) | Each entry represents a distinct assistance |

*Relation*

Resident (Email, FName, LName, Building, Apt #, Phone)

Primary key: Email

Participant (Participant ID, Name, Adult/Kid, Email, Special Assistance)

Primary key: Participant ID

1. **Test Plans**

***7.1. Objectives***

*7.1.1. Purpose*

This document describes the plan to test the prototype of the Event Registration System. This Test Plan supports the following objectives:

* Describe the testing strategies to be used.
* Identify testing resources and estimation of testing efforts and schedule.
* Identify testing deliverables.

*7.1.2. Scope*

The Event Registration System prototype is to provide an online form to register for an event. Its three main features will be tested including:

* Event registration
* Event cancellation
* Slot availability

***7.2. Requirements for the Test***

The list of requirements show what will be tested.

*7.2.1. Data and database integrity testing*

* Verify access to Registrant and Participant database.
* Verify lockout during database updates.
* Verify correct updates of data.

*7.2.2. User interface testing*

* Verify ease of navigation when using the tab through the fields.
* Verify ease of use in the design.

*7.2.3. Performance testing*

* Verify the loading time from clicking Register to successful registration.
* Verify the loading time to unhide optional fields after choosing Cancel option.
* Verify the loading time from clicking Cencel to successful cancellation.

***7.3. Resources***

*7.3.1. Human Resource*

|  |  |
| --- | --- |
| **Name** | **Responsibilites** |
| Huyen Nguyen | * Generate test plan * Evaluate effectiveness of test efforts |
| Quyen Nguyen | * Generate test cases * Provide technical direction |
| Daniel Sethuraman | * Execute test cases * Log results * Recover from errors |
| Vignesh Dhanabal | * Administer test database * Evaluate test completion criteria |

*7.3.2. System*

* Windows 10/XP/Vista, SQL Server
* Tools: manual testing

*7.3.3. Project Milestones*

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort** | **Start date** | **End date** |
| Test Plan | 8 man-hour | 4/8/16 | 4/8/16 |
| Test Design | 32 man-hour | 4/11/16 | 4/14/16 |
| Test Execution | 32 man-hour | 4/15/16 | 4/20/16 |
| Test Evaluation | 4 man-hour | 4/21/16 | 4/21/16 |

*7.3.4. Deliverables*

* Test cases and corresponding results
* Test data

***7.4. Dependencies/ Risks***

Testing and fixing errors may be time consuming and the project milestones may not be met as planned.

***7.5. Test cases***

|  |  |
| --- | --- |
| Test Case ID | 1 |
| Test Description | Event registration |
| Revision History | 4/7/2016 – test case creation |
| Function to be tested | Event registration |
| Environment | Window 10 |
| Test execution | Enter mandatory information: registrant’s first name, last name, building and apartment number, email, number of adults, number of children, and one adult’s name and one child’s name  Enter optional information: registrant’s phone, other adult’s name and child’s name of there is more than one participant in each category |
| Expected result | If all mandatory fields are filled all the results should be displayed and registration is successfully completed, registered information should be sent to the database |
| Actual result | To be tested |

|  |  |
| --- | --- |
| Test Case ID | 2 |
| Test Description | Event cancellation |
| Revision History | 4/7/2016 – test case creation |
| Function to be tested | Event cancellation |
| Environment | Window 10 |
| Test execution | Click Cancel button, enter mandatory fields, and submit Cancel again |
| Expected result | All optional fields will be hidden after the Cancel button is hit.  If all mandatory fields are filled, cancellation is successfully completed, cancelled information should be sent to the database |
| Actual result | To be tested |

|  |  |
| --- | --- |
| Test Case ID | 3 |
| Test Description | Slot availability |
| Revision History | 4/7/2016 – test case creation |
| Function to be tested | Running number of slots left |
| Environment | Window 10 |
| Test execution | Enter the maximum number of available slots and generate a successful registration as well as successful cancellation |
| Expected result | The number of slots left should be updated after each successful registration and cancellation. It is equal to the maximum number of slots minus the number of adults and the number of children after a successful registration, and equal to the current number of slots plus the number of adults and children after a successful cancellation |
| Actual result | To be tested |