

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2022

Rakshak (Hire a Friend)

Dinesh Sundaram

Follow this and additional works at: <https://opus.govst.edu/capstones>



Part of the [Technology and Innovation Commons](#)

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

RAKSHAK (HIRE A FRIEND)

By

Dinesh Sundaram

Bachelor in Electronic Science, University of Madras, 2005

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2022

ABSTRACT

In today's busy world, people or families always needs support through some or other means to move forward with their daily routine life. For example: On a daily life routine, we may have come across situations where we may need skilled resources to help us for "Ex: House Keeping", "Shifting apartments", "Plumbing or Electrical Appliance Maintenance", "Wall Painting...etc". But, while we try to contact the resources we need (contacts through several sources), we often come across many challenges which are as documented below:

- Fit for purpose (Does the resource has relevant skill set).
- Manual effort is always required. We need to connect with neighbors or go through search engines as we don't have a quick medium or channel to support.
- Delay or lack of communication channel. We may not be able to help the needy on-time.
- Finding resources within an accessible & practically feasible location.

Considering the above defined problem statement, we have developed a web application known as "**Rakshak**" (i.e.,) **Hire a Friend** which provides a technology platform to help and facilitate people / families with the required capabilities to find the skilled resources they need. "Rakshak" is going to be a communication channel & centralized platform which bridges the gap between "**Needy Customers**" & "**Skilled Professionals**".

With our current presence in Industry 4.0 where we consistently see technology evolve progressively starting with cyber-physical systems, IOT's, Cognitive computing, it's always a known fact that 100% automation "of all day-to- day tasks" is never possible or it could become possible over a decade (which means there lies significant time). We still need "Skilled Human Resources" to accomplish certain basic tasks which don't have an equivalent substitute. So, technology platforms built with thought processes like "Rakshak" will not only help digitization, but also purposefully help bridge the gap between "what we need" vs "finding the right fit for purpose".

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects.....	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	2
1.5	Definitions and Acronyms.....	2
2	<i>Project Technical Description</i>	2
2.1	Application Architecture	3
2.2	Application Information flows.....	5
2.3	Interactions with other Applications.....	15
2.4	Capabilities	15
2.5	Risk Assessment and Management.....	16
3	<i>Project Requirements</i>	16
3.1	Identification of Requirements	16
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P).....	17
3.3	Security and Fraud Prevention.....	18
3.4	Release and Transition Plan.....	18
4	<i>Project Design Description</i>	18
5	<i>Internal/external Interface Impacts and Specification</i>	20
6	<i>Design Units Impacts</i>	21
6.1	Functional Area A/Design Unit A	21
6.1.1	<i>Functional Overview</i>	21
6.1.2	<i>Impacts</i>	23
6.1.3	<i>Requirements</i>	23
6.2	Functional Area A/Design Unit A	23
6.2.1	<i>Functional Overview</i>	23
6.2.2	<i>Impacts</i>	23
6.2.3	<i>Requirements</i>	24
7	<i>Acknowledgements</i>	24
8	<i>References</i>	24

1 Project Description

We have developed a web application known as “**Rakshak**” (i.e.,) **Hire a Friend** which provides a technology platform to help and facilitate people / families with the required capabilities to find the skilled resources they need. “**Rakshak**” is going to be a **centralized platform** which bridges the gap between “Needy Customers” & “Skilled Professionals”.

In our present information technology space, product individuality & the value it offers to its customers define the success or failure of any business model. Again, the product’s individuality or its “features” should be aligned with the “value” which is being offered to its customers. In those words, we as a team also have thought through in bringing valuable core features which could help not only to establish a trustworthy branding, but again more focused on the quality of service being delivered.

- Transparency of a Service Request between Admin, Customers & Professionals
- Manage/Control the workflow of each request through procedural/logical steps.
- Cancel or Reject a Service Request through authorized roles & in applicable phases of workflow.
- Controlled Service request closure through OTP (One Time Passcode) feature.
- Facilitate filters and dashboard to provide visibility among stakeholders.

1.1 Competitive Information

The first & foremost thought which reached our mind while coming up with the idea about “Rakshak” platform is our own family and especially how much each family head (Parents) take ownership of responsibilities and how much struggle they go through every day while raising their family. Again, those days we didn’t have mobile or computing capabilities as we have today. And, we have seen them struggling to find the right resources to network with or approach them with confidence because “Problems” are never ending and, it’s about spending their hard-earned money towards an essential cause. So, we strongly believe that “Rakshak” will not only be considered as “bridge a gap infrastructure” but would also serve as source of light to bring those “hard working professionals” and helps establish a meaningful self-identity and gives them opportunity to stay connected with world in this growing technological space.

We could co-relate the concept behind “Rakshak” with Home Services applications which are presently being offered by different vendor platforms (**Example: Urban Company, House Job**). Home services applications will continue to evolve in future especially because of the convenience and accessibility which this platform facilitates to end users. As per a recent study, the demand for online home services applications is expected to evolve with an annual growth rate of 17% in-between 2022 till 2026.

1.2 Relationship to Other Applications/Projects

The scope of this idea “Rakshak” could be referenced with some of the existing solutions providing “Home Services”. But we have tried differentiating by providing our own thought process (Ex: OTP closure, Filters, Transparency on Workflow status) which would define our uniqueness to our developed product. Also, we have facilitated Administrators with the capability to Add more services or Service Locations though suitable interfaces which could help business to scale considering the broader spectrum of opportunities based on future demand.

1.3 Assumptions and Dependencies

- Present solution would use our local environment for the purpose of development and testing effort.
- For testing some of the intermediate functional flows (Example:Data visualization in Dashboard), needed workaround approaches would be used (rather than waiting for the actual functional interface)
- We are using mocked/non-realistic Email ID’s & Address information which would serve the purpose of development and testing effort.
- Sequence & Priority Order in which requirements are being accomplished could change as it’s agile development.
- Service professionals could be hired by Admin, or the platform is Open for skilled professionals to register and it’s the decision of Admin to assign suitable service professionals to customer-initiated requests.
- Each service professionals could comprise a diversified skillset (Plumbing, Cleaning, Cooking...etc).
- The service request closure or cancellation requests would be initiated through phone calls or emails through Admin or Professionals.

1.4 Future Enhancements

We presently do NOT have any enhancements planned for this solution as our focus was to accomplish the tasks which we initially planned, and we had a definitive goal in-terms of timelines. At the same time, we also adhere to the thought process which states “Change is the only constant”. If this application needs to be implemented on a real time scenario, we could add more features which could help improve the communication between Customer and the Professional. Especially, we could also extend our OTP concept to the Rejection & Cancel flows. Also, capabilities to facilitate transparency around the feedback loop.

1.5 Definitions and Acronyms

Below documented are some of the definitions and acronyms which we have used in our project.

ASP.NET – Active Server Pages

OTP – One Time Passcode

SQL – Structured Query Language

HTML – Hyper Text Markup Language

Admin – Referencing Administrator

2 Project Technical Description

We have used “ASP.NET Core 3.1 with Entity framework” for our solution development, especially because its open source and we have a good community and knowledge base for references. Moreover, we know it supports different platforms like Windows & Mac. In-addition .NET Core could also support integration with other frameworks which could be a good capability to consider incase if the application requires future scaling or enhanced features or like integration with different vendor systems. Also, its ability to utilize in-built hosting platforms like IIS which makes it different and unique from other programming tools (Example: Using TomCat Apache server to host a JSP built application). Below documented is a set of tools and framework which we have used to build this application.

- ASP.NET Core 3.1 with Entity Framework, Bootstrap
- Microsoft Visual Studio 2022 Community Edition
- Microsoft SQL Server Management Studio
- IE Edge Browser (Version: 107.0.1418.56)
- Chrome Browser (Version: 107.0.5304.107)

User roles for this solution would comprise the following stakeholders & technical capabilities.

Customers: Users with roles “Customers” will have the capability to create customer accounts and create service requests (Example: House Keeping, Plumbing, Electrical App Maintenance...etc). Customers will be able to update their profile, view the workflow status of the service request or even cancel the created service request. In-addition the closure flow of Service request will be controlled by customer as they would hold the OTP for the service professional to technically consider the request as closed.

Professionals: Users with roles “Professionals” will have the capability to act upon a service request (as when assigned by the administrator). Professionals will be able to move forward with the workflow status of the service request or even reject the assigned service request (in case if customer requests for it).

Admin: User with roles “Admin” would have the capability to manage service locations (add/edit) and services (add/edit) based on business needs. Admins would have the authority to assign a service professional to a customer-initiated service request or even Cancel the customer-initiated service request (in case of resource limitations or customer request). Admins would be facilitated with the Dashboard view along with different filters to manage the booking status of requests which were initiated by different customers.

2.1 Application Architecture

Core Modules developed for this project work is demonstrated in below Figure 2.1.1.

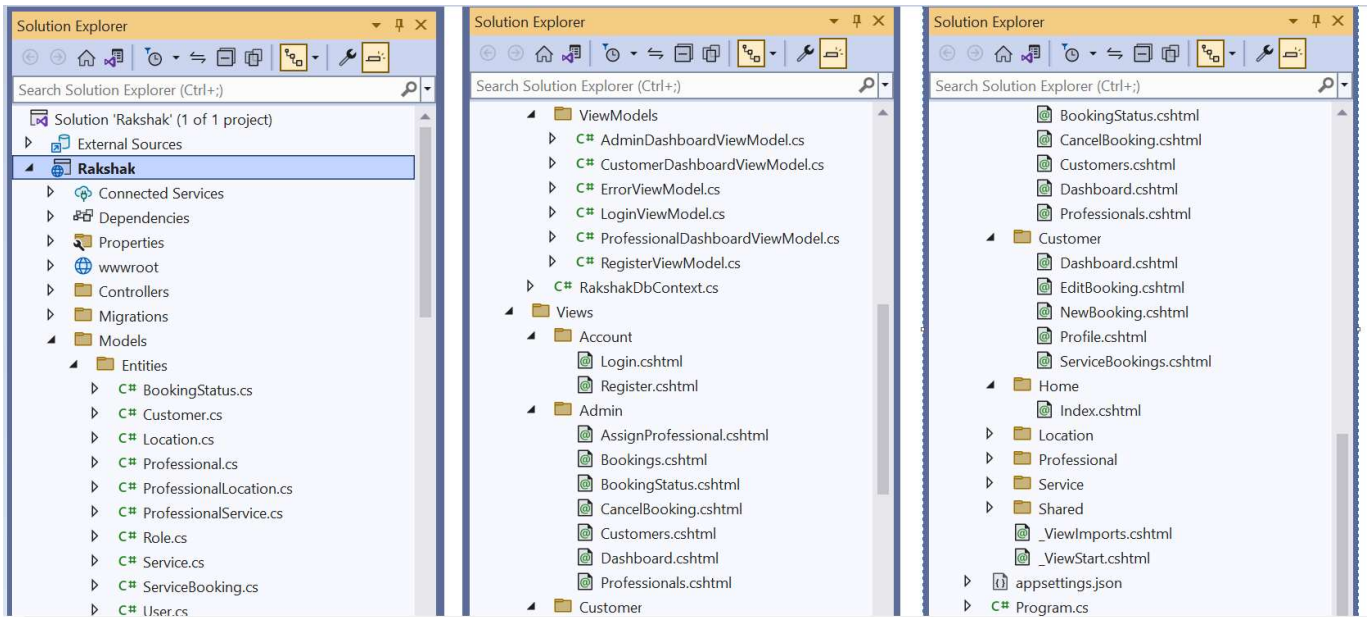


Figure 2.1.1 - Core Application Modules in Rakshak Portal:

Connection string to establish database connectivity with Rakshak application is demonstrated in below Figure 2.1.2.

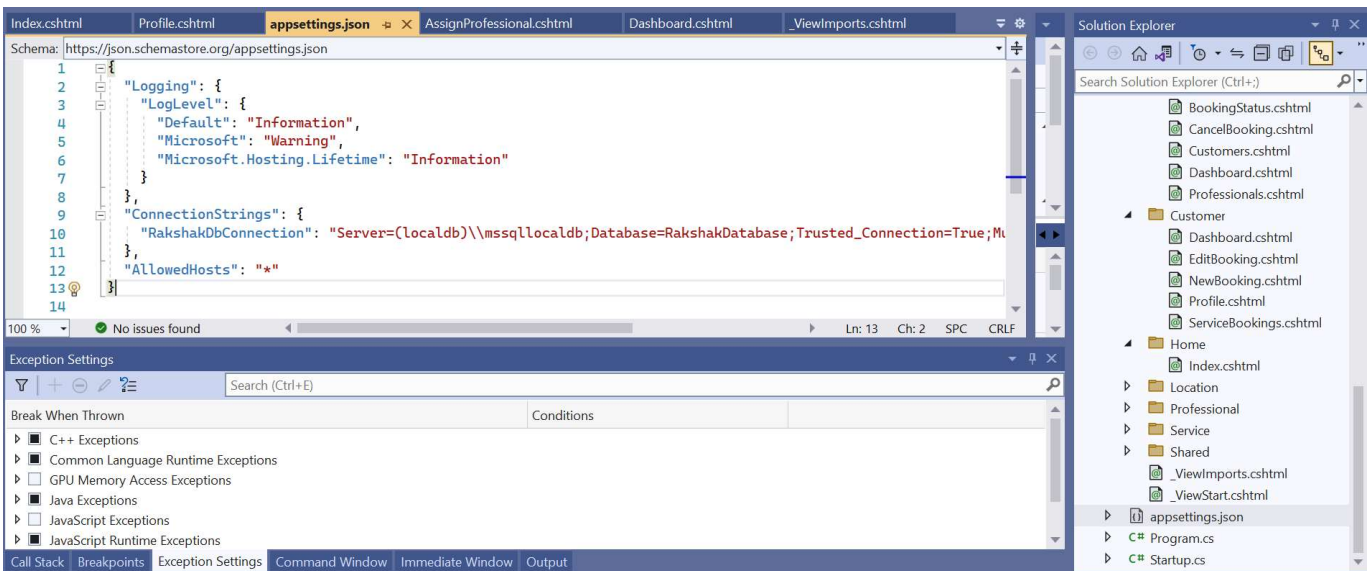


Figure 2.1.2 - Database Connection (appsettings.json):

Application configuration settings which call the required environmental dependencies is demonstrated in below Figure 2.1.3.

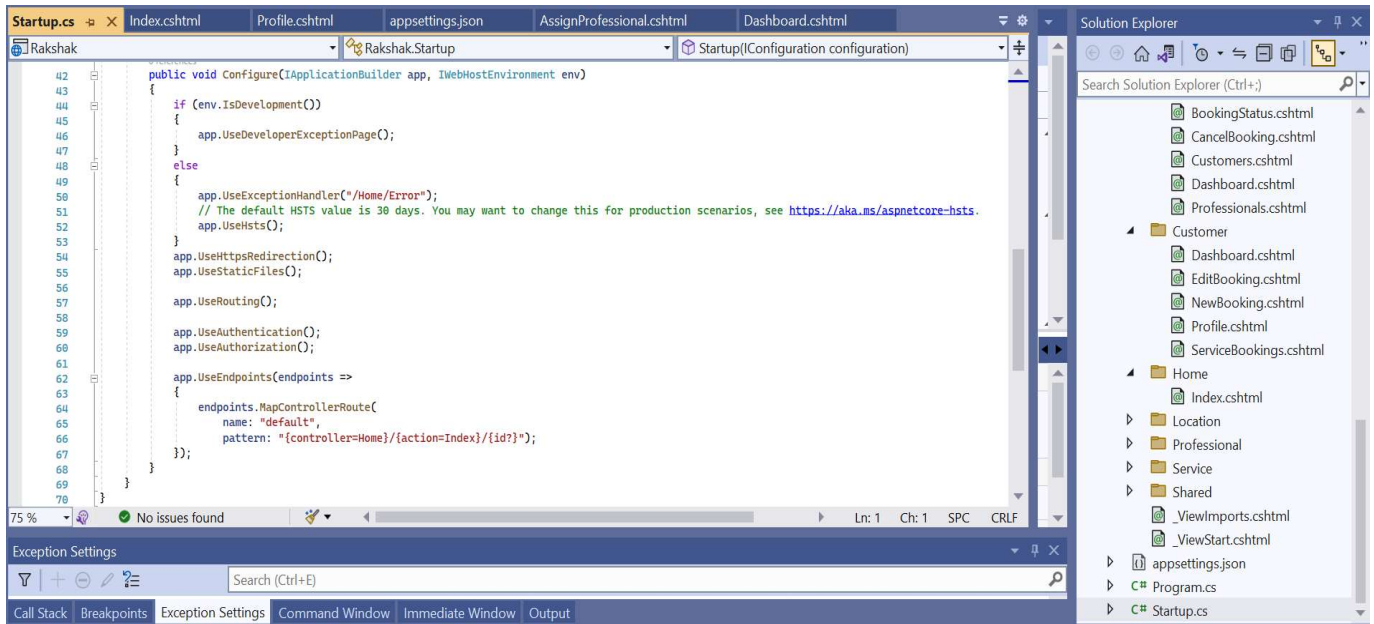


Figure 2.1.3 - Application Configuration Settings:

The relationship between different entities in the application database is illustrated in below Figure 2.1.4.

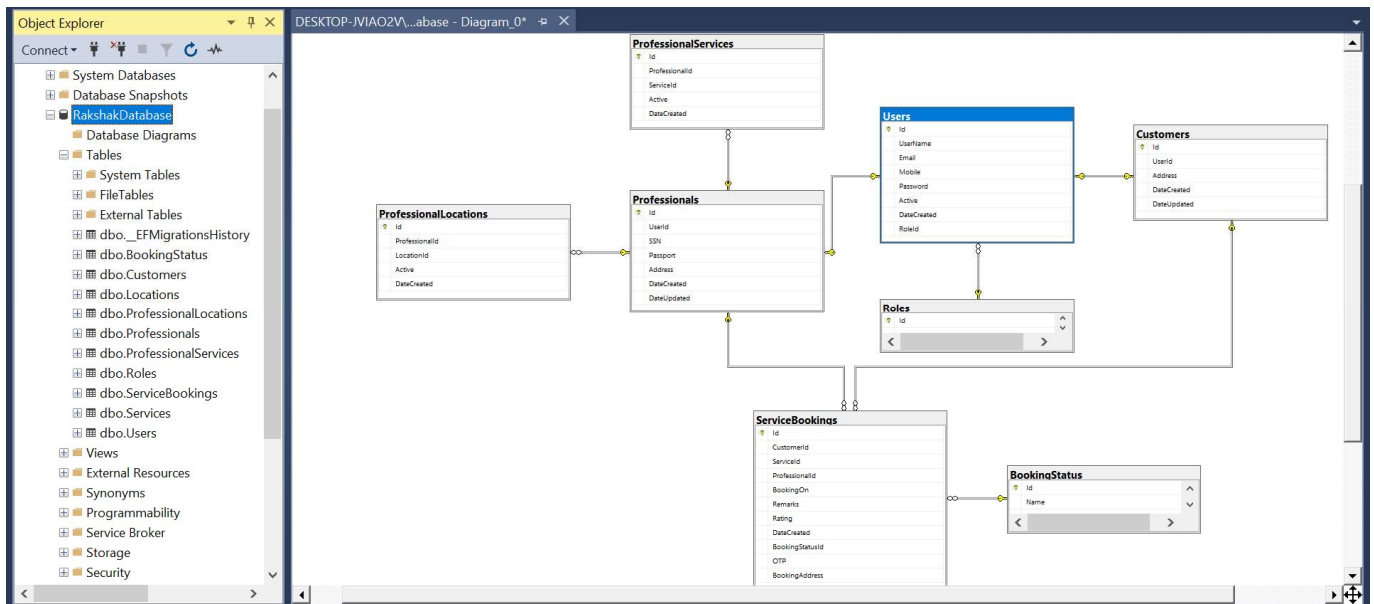


Figure 2.1.4 - Database ERD:

The overall “Rakshak” solution core functional workflow is as explained in the below Figure 2.1.5.

- Transparency of a Service Request between Admin, Customers & Professionals.
- Customers register their profile in “Rakshak” portal.
- Registered Customer logs into “Rakshak” portal and “Books a new service” (Example: Plumbing, Cleaning...etc)
- Admin views the bookings initiated by customer through Booking Status screen.
- Admin Assigns a suitable Professional to the Service Bookings.
- Professional carries forward with the Service Booking Workflow (In-Progress, Completed or Rejected).
- Post-Completion of assigned task, the professional co-ordinates with the customers, pursues the OTP and closes the service request in Rakshak Portal.

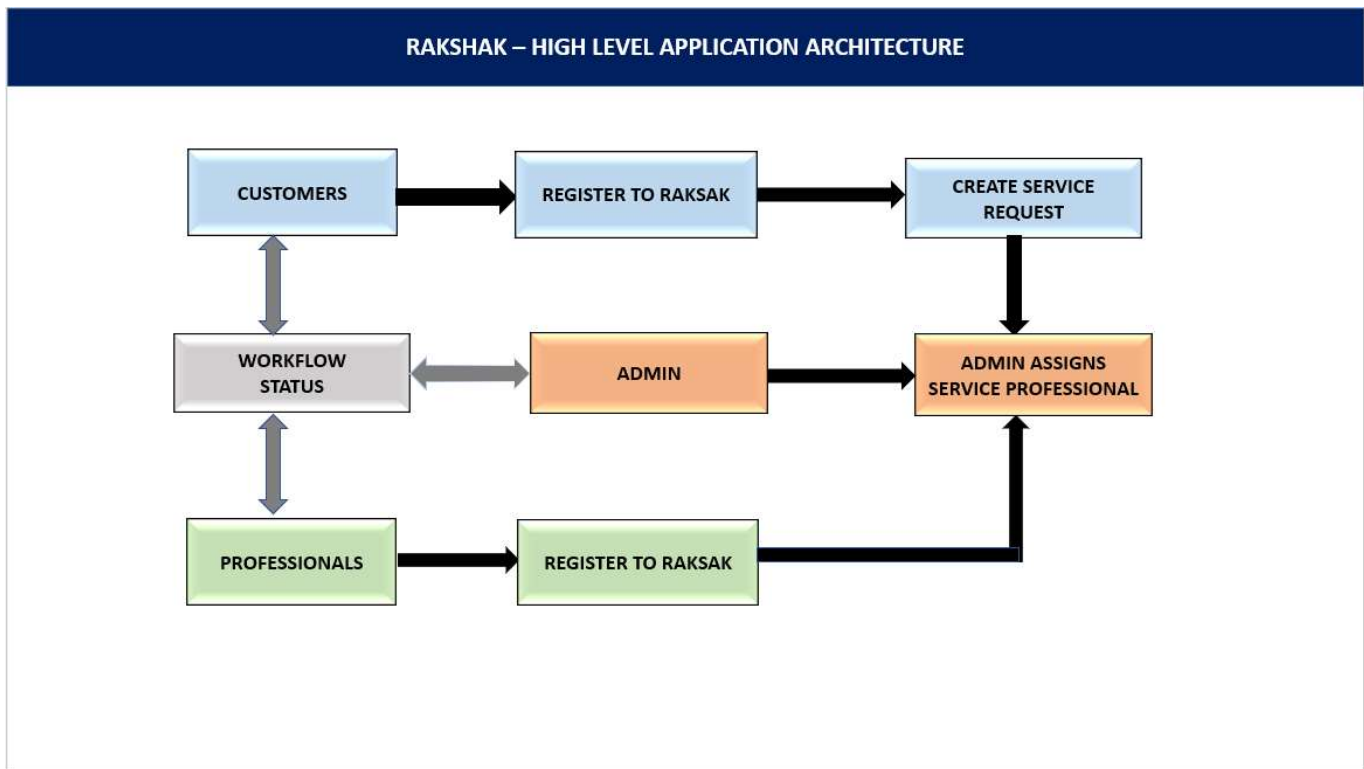


Figure 2.1.5 - Application Architecture of Rakshak:

2.2 Application Information flows

The core information flow in the form of “Workflow” between the three stakeholders (Admin, Customer & Professional) is all illustrated in below Figure 2.2.1.

- Customer creates a new service request and each booking request comprise a “Booking ID#”.
- Each Booking ID traverses through the workflow (when it satisfies the below defined criteria per our business logic requirements).
- While a Booking ID is in “Open” status, the Customer or the Admin has the authority to Cancel the Booking.
- While a professional is assigned, the corresponding professional will have the capability to move the “Booking ID#” status from “Open” to “Work in Progress”.
- In-addition the professional has the authority to Reject the Booking with proper reason (either requested by customer or due to unavailability).
- Once when the assigned task is completed, the professional moves the Booking ID# from “Work in-progress” to “Work Completed”.

- During this stage transition while moving from “Work in-progress” to “Work Completed”, the customer will have OTP displayed in his Bookings view.
- Professional can co-ordinate with corresponding Customer, get the OTP for the corresponding booking ID and can move the Booking Status to “Closed” (Only if customer given OTP matches).

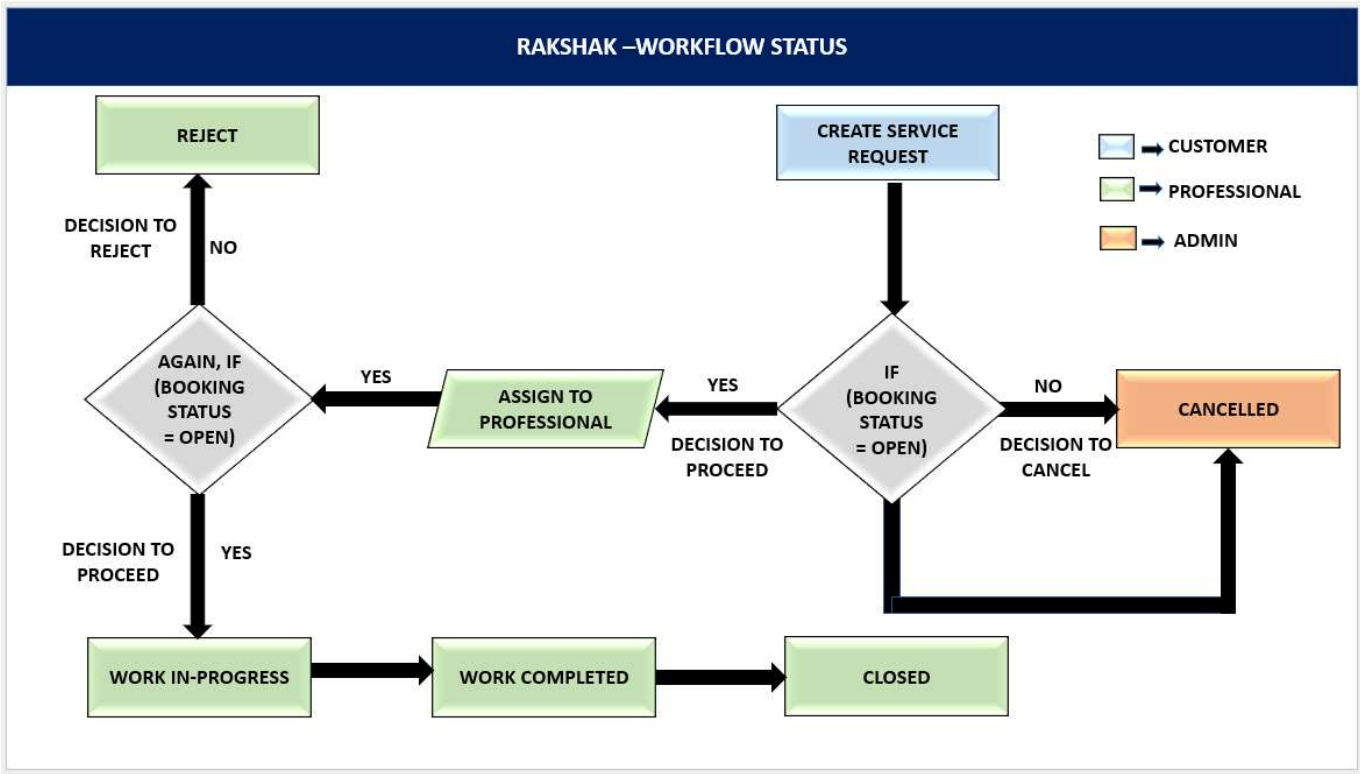


Figure 2.2.1 - Data flow demonstrating workflow status:

- Whenever, any user launches the Rakshak application portal, the “Home” user interface illustrated in below Figure 2.2.2 will be presented to the user.

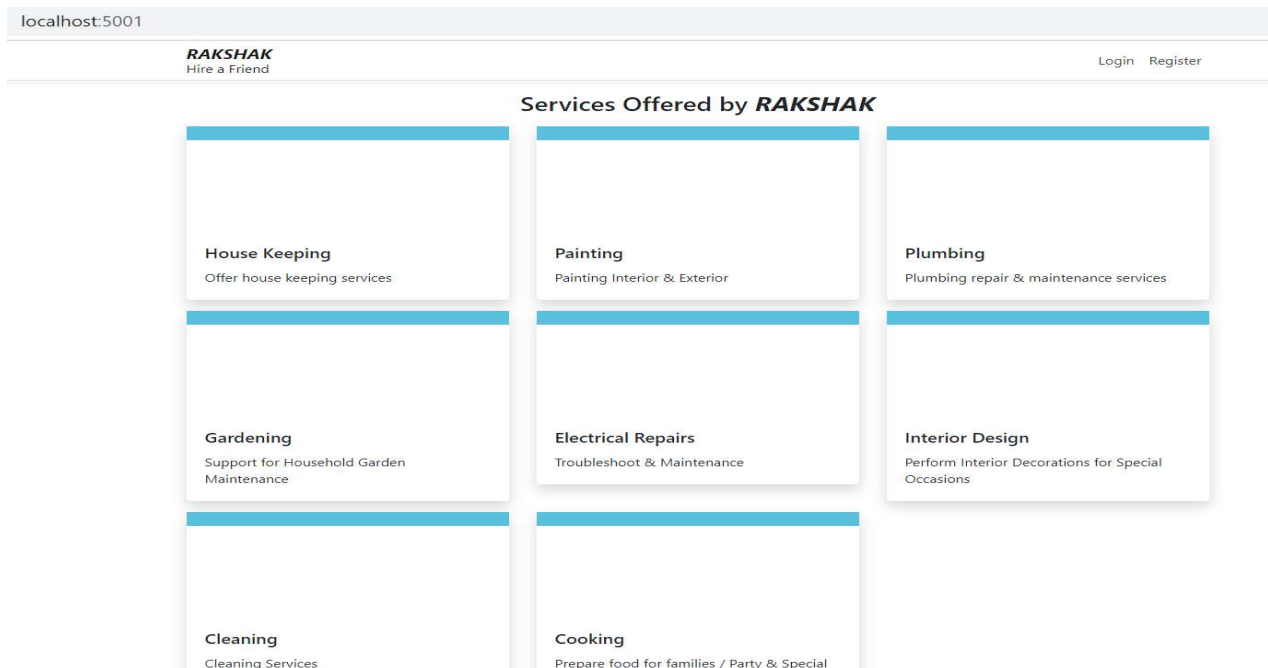


Figure 2.2.2 - Home Screen User Interface:

- Users can register to Rakshak application as “customer” through the user interface illustrated in Figure 2.2.3.

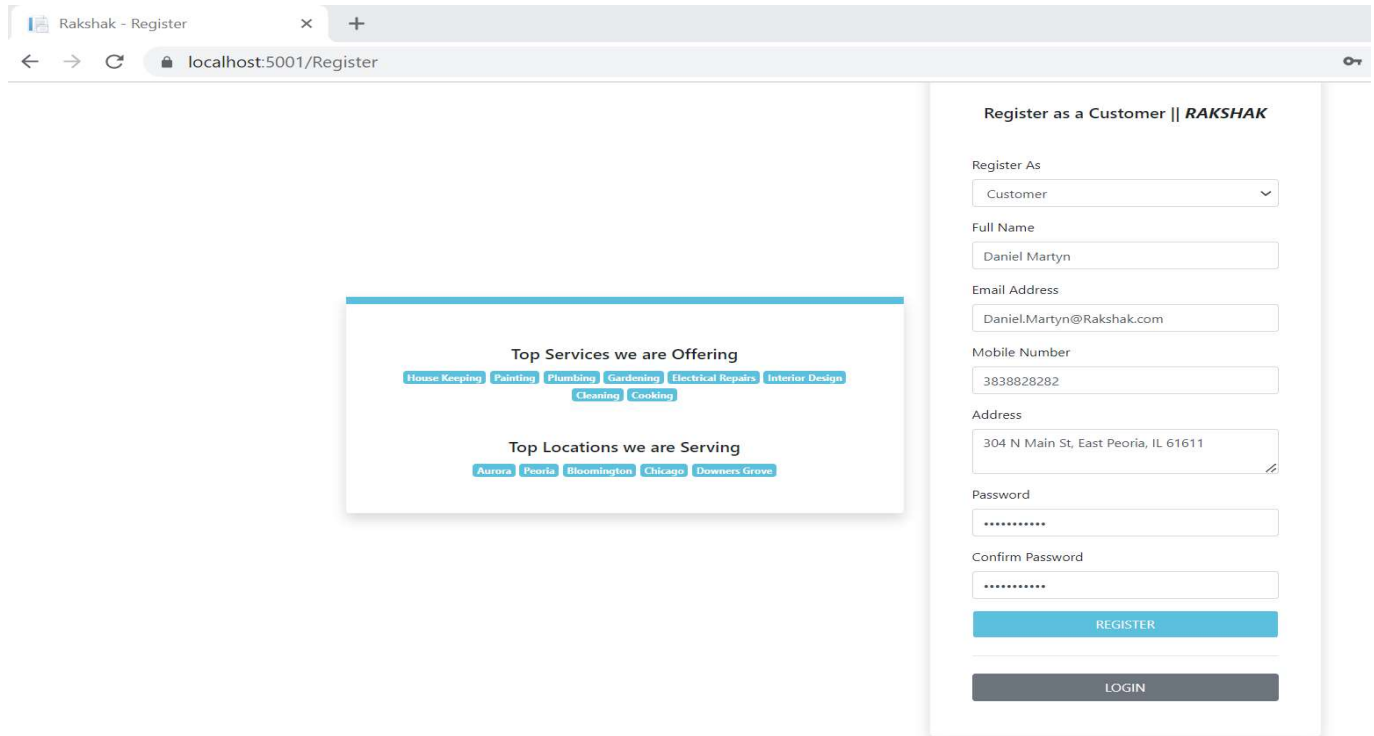


Figure 2.2.3 - Register as Customer User Interface:

- Registered users can login to application through the user interface illustrated in Figure 2.2.4.

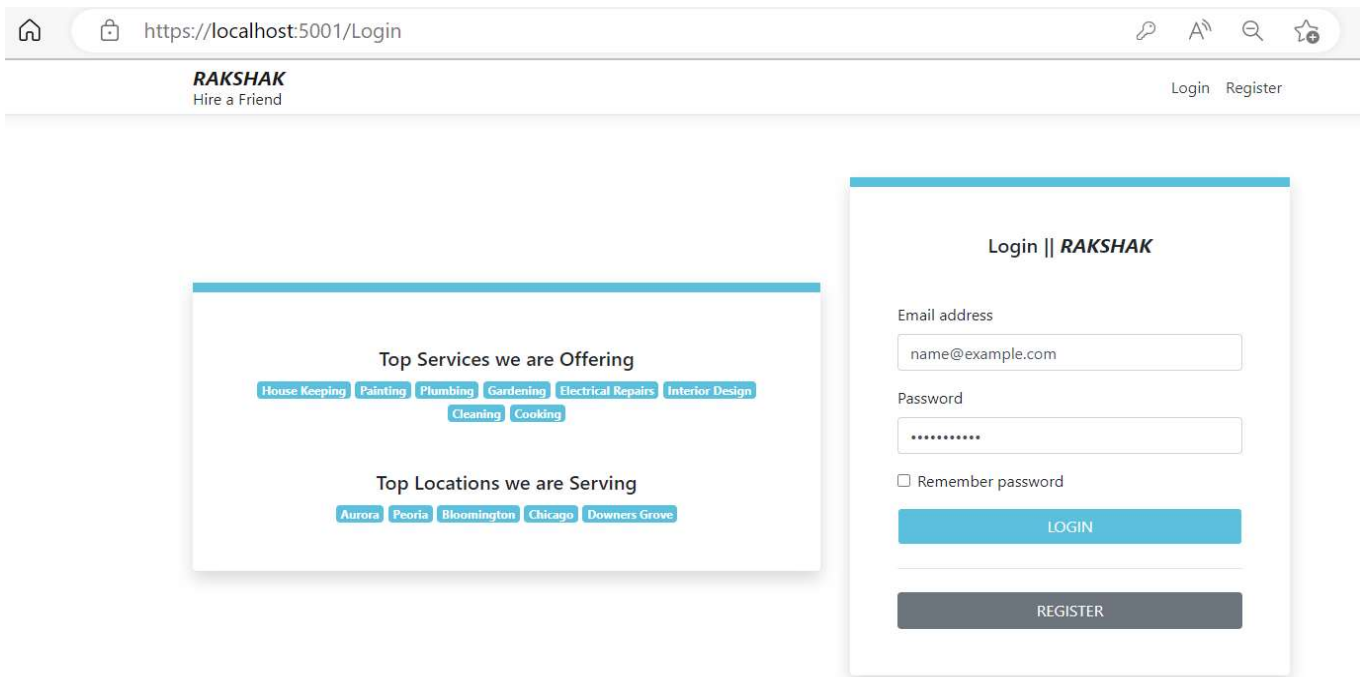


Figure 2.2.4 - Login Screen User Interface:

- The dashboard user interface presented to “Customer” after logging into application is illustrated in Figure 2.2.5.

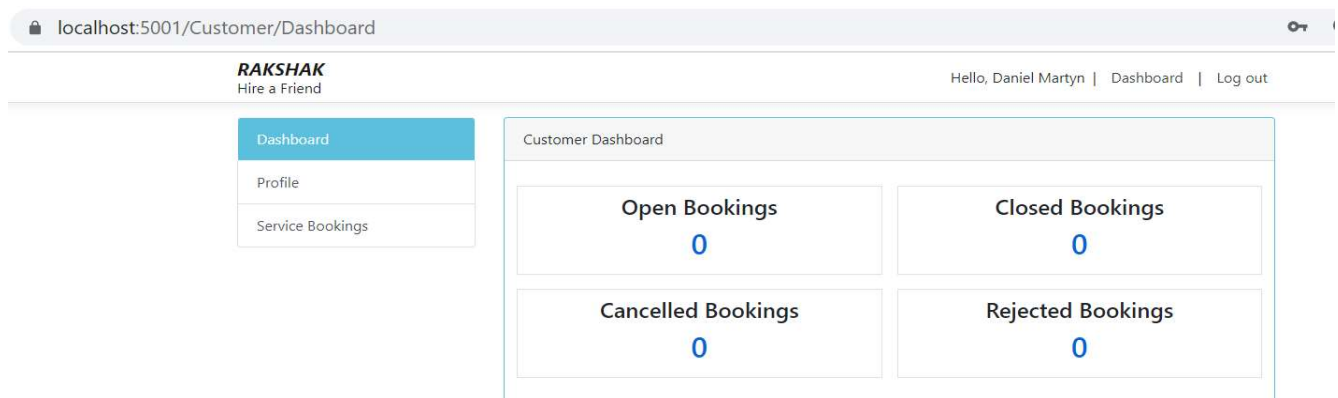


Figure 2.2.5 – Customer Dashboard User Interface:

- Registered customers can update their profile details using the user interface illustrated in Figure 2.2.6

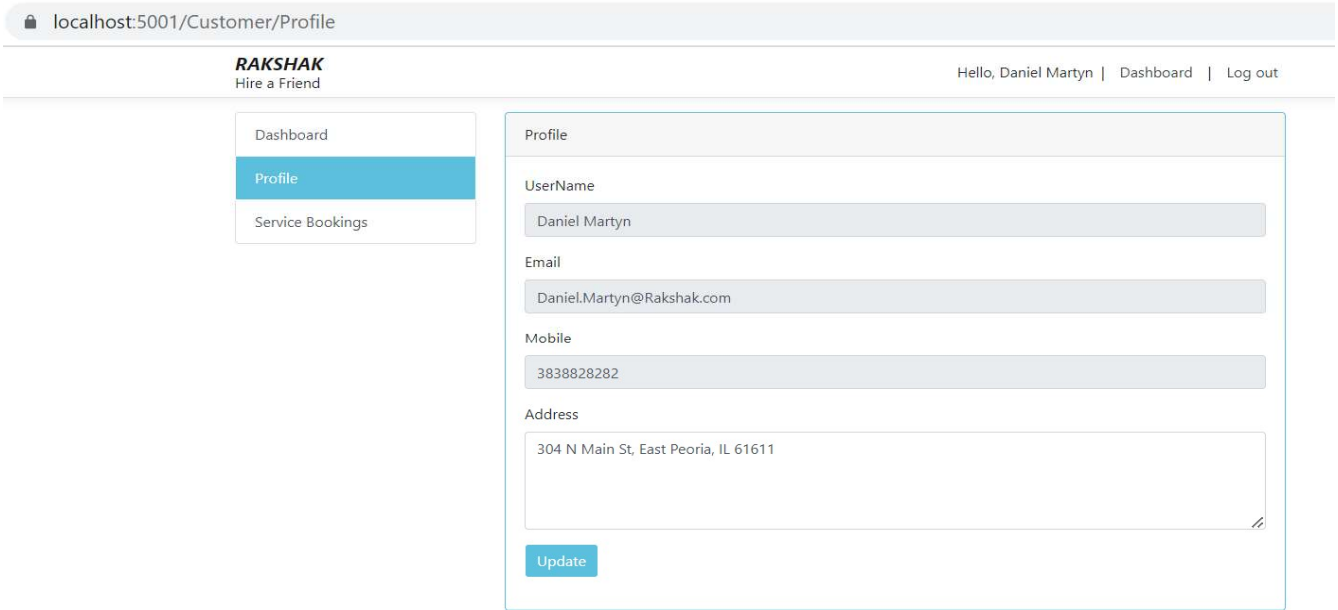


Figure 2.2.6 - Update Customer Profile:

- Registered customers can “Book a New Service” and view their status through the user interfaces illustrated in Figure 2.2.7 & Figure 2.2.8.

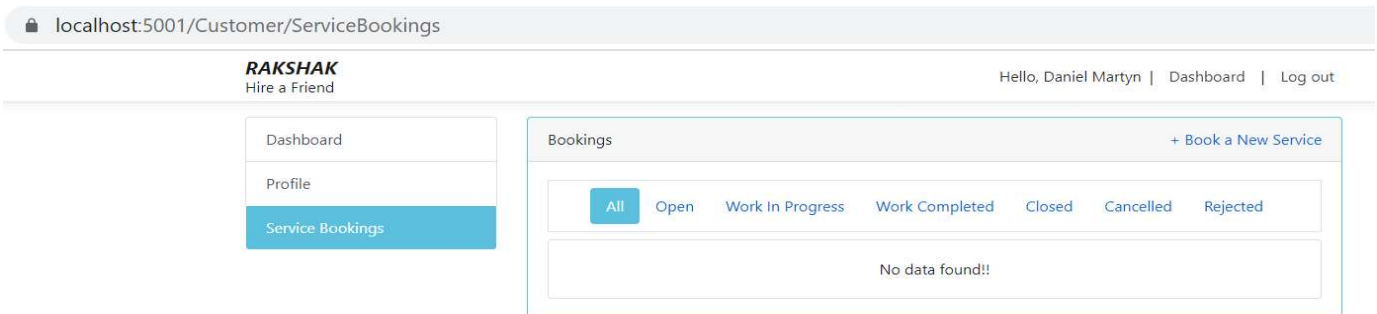


Figure 2.2.7 - Customer Service Bookings to display status:

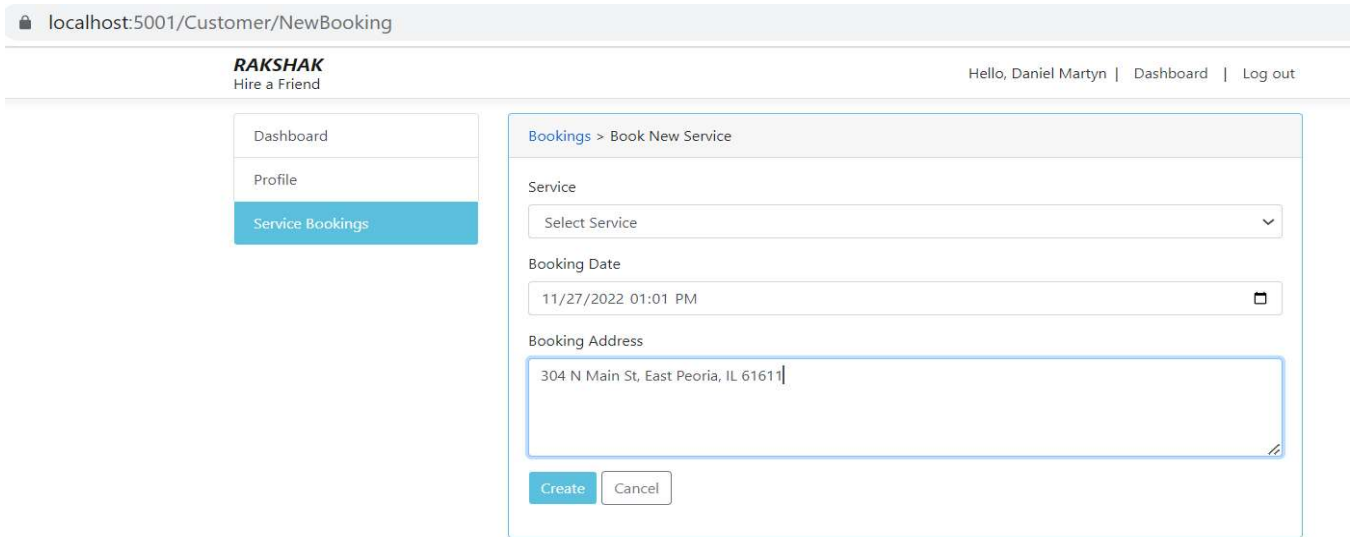


Figure 2.2.8 - Book New Service Request User Interface:

- Service request created by customers will be displayed in Service Bookings user interface illustrated in Figure 2.2.9

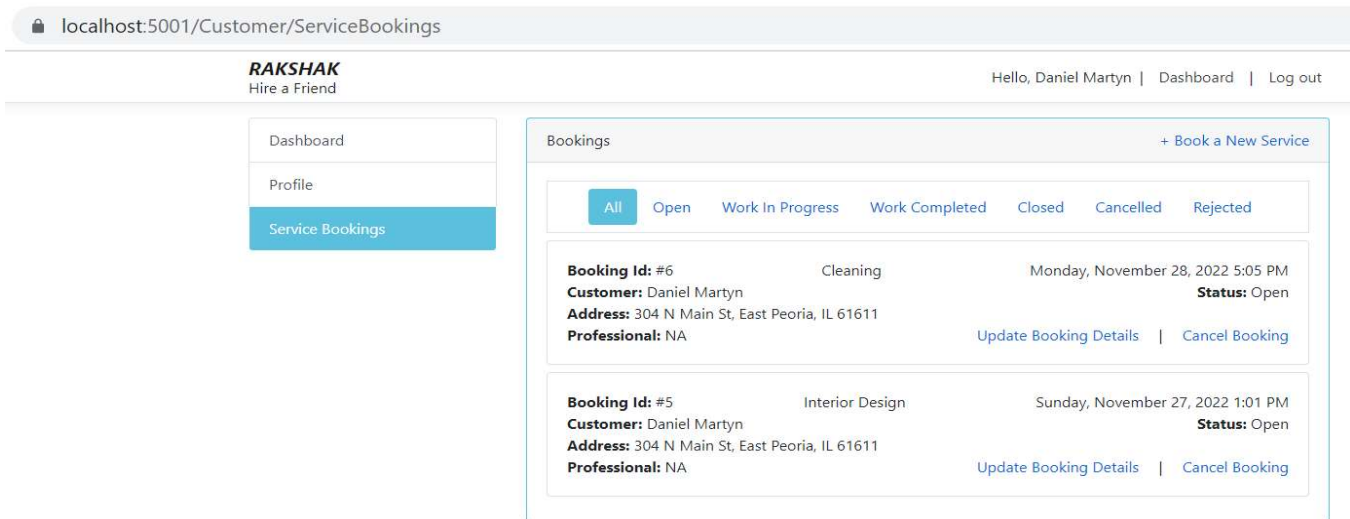


Figure 2.2.9 - Customer Service Bookings View:

- Service request created by customers can be cancelled through the Cancel Booking user interface illustrated in Figure 2.2.10

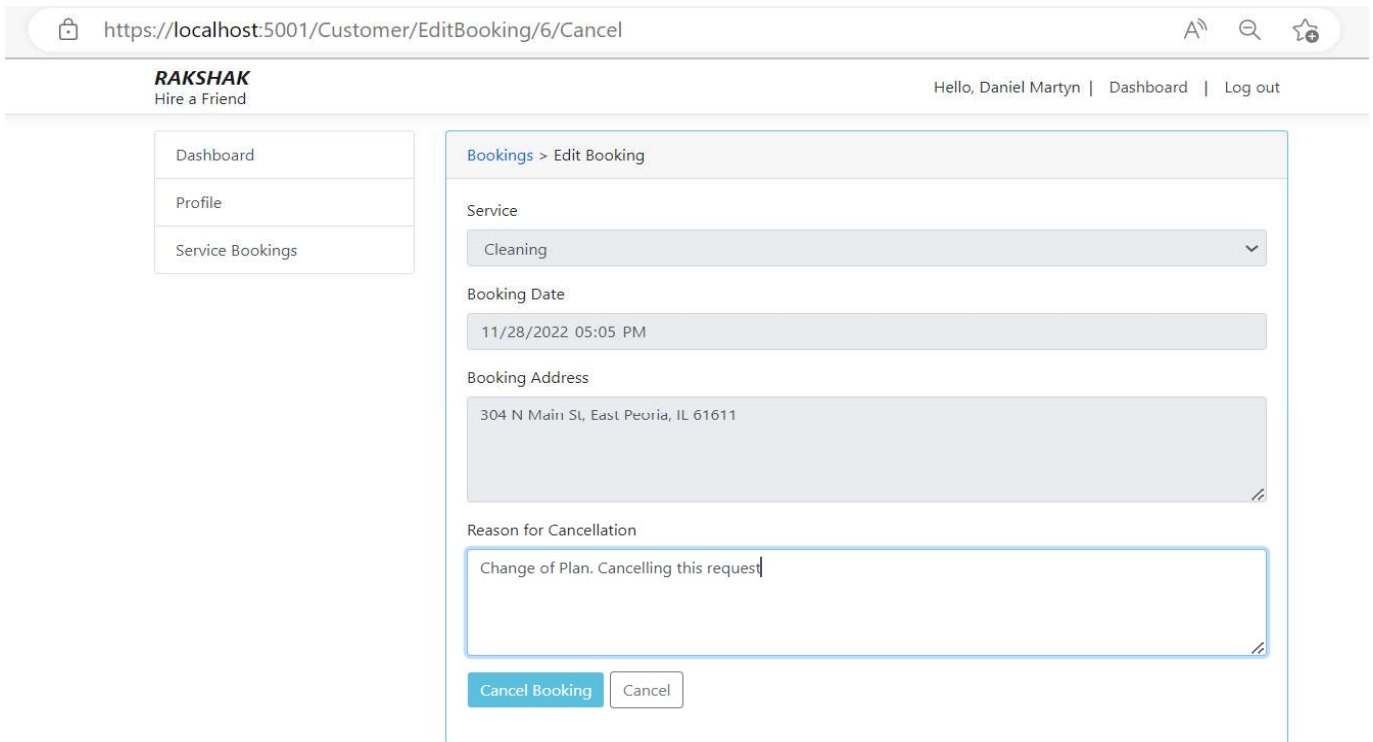


Figure 2.2.10 - Customer Cancel Bookings View:

- Professionals will be able to register into Rakshak application through the user interface illustrated in Figure 2.2.11.

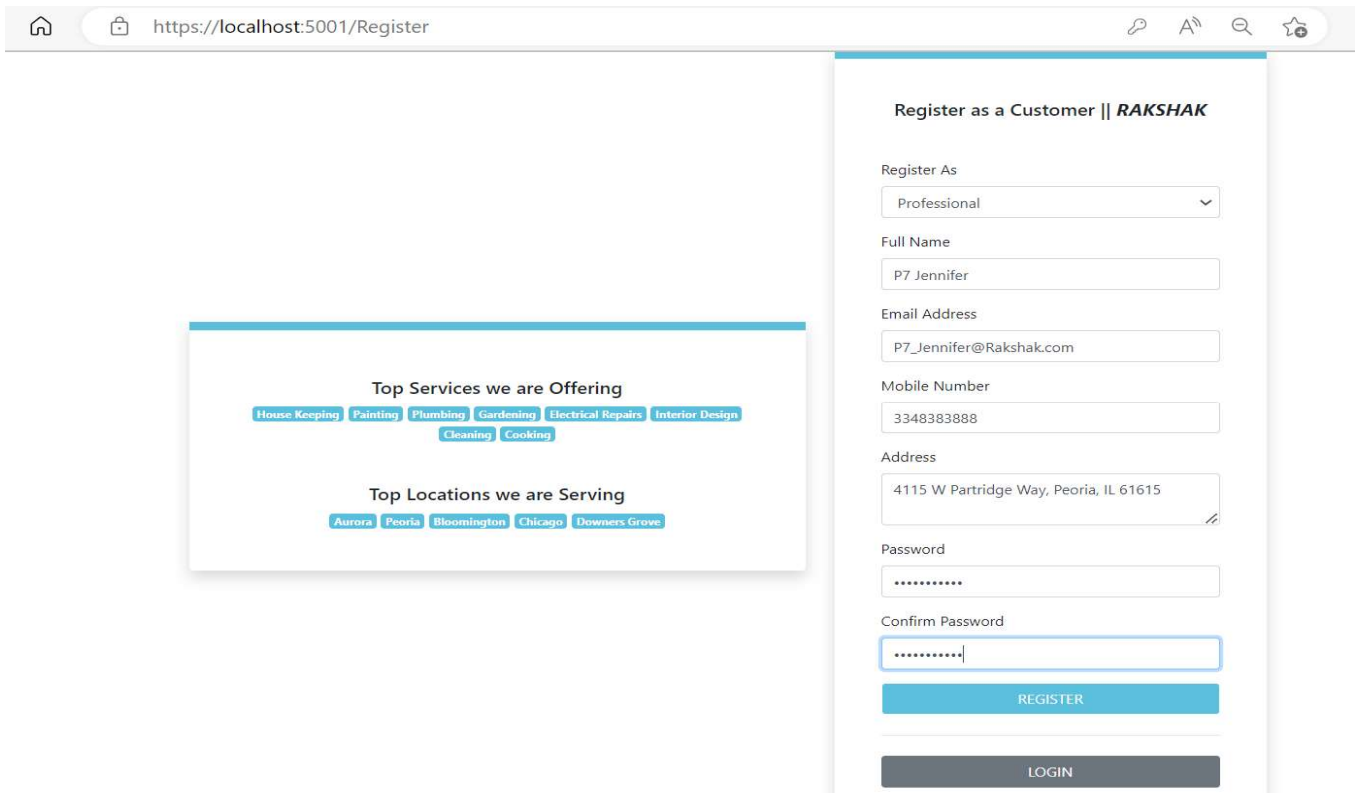


Figure 2.2.11 - Register as Professional:

- Admin users will be able to view the bookings initiated by “Customers” through the user interface illustrated in Figure 2.2.12.

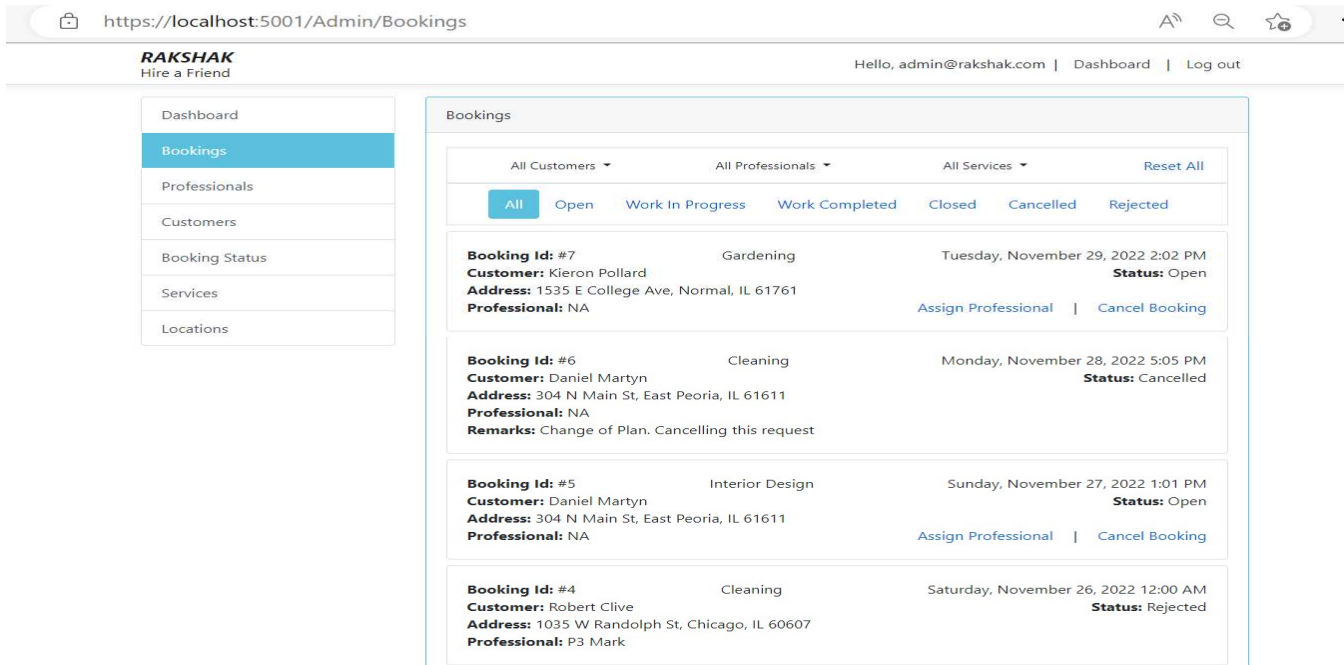


Figure 2.2.12 - Admin Bookings View:

- Admin users will be able to assign a professional to “Customer Booking” through the user interface illustrated in Figure 2.2.13.

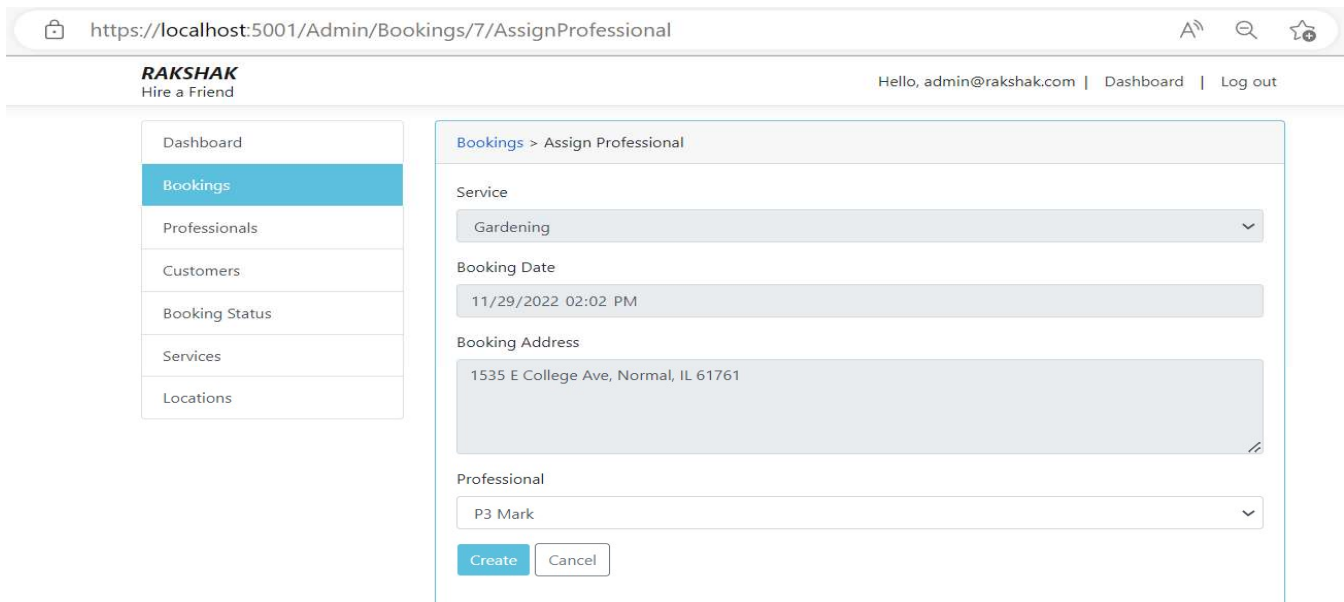


Figure 2.2.13 - Admin Assigns Professional to Customer Bookings:

- “Professional” will be able to move forward with workflow status through the user interface illustrated in Figure 2.2.14.

https://localhost:5001/Professional/ChangeStatus/7/2

RAKSHAK
Hire a Friend

Hello, P3 Mark | Dashboard | Log out

Dashboard
Service Bookings

Bookings > Change Booking Status

Service
Gardening

Booking Date
11/29/2022 02:02 PM

Booking Address
1535 E College Ave, Normal, IL 61761

Booking Status
Work In Progress

Create Cancel

Figure 2.2.14 - Professional – Moving Workflow status – Work in Progress

- “Professional” will be able to reject assigned customer bookings through the user interface illustrated in Figure 2.2.15.

https://localhost:5001/Professional/ChangeStatus/7/6

RAKSHAK
Hire a Friend

Hello, P3 Mark | Dashboard | Log out

Dashboard
Service Bookings

Bookings > Change Booking Status

Service
Gardening

Booking Date
11/29/2022 02:02 PM

Booking Address
1535 E College Ave, Normal, IL 61761

Booking Status
Rejected

Reject Reason
Customer informed to reject.

Create Cancel

Figure 2.2.15 - Professional – Moving Workflow status – Rejected:

- “Customers” will be facilitated with “OTP” to control the closure of service bookings as illustrated in Figure 2.2.16.

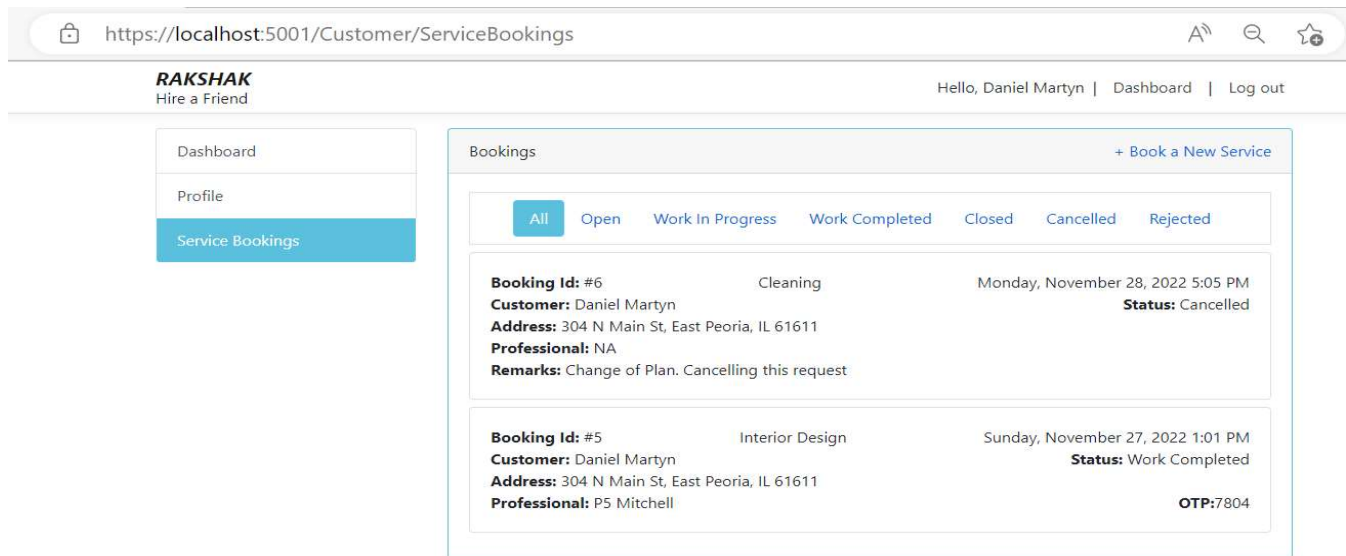


Figure 2.2.16 - Customer Service Bookings - Displaying OTP (Work Completed):

- “Professionals” will be able to close the assigned service bookings upon completion as illustrated in Figure 2.2.17.

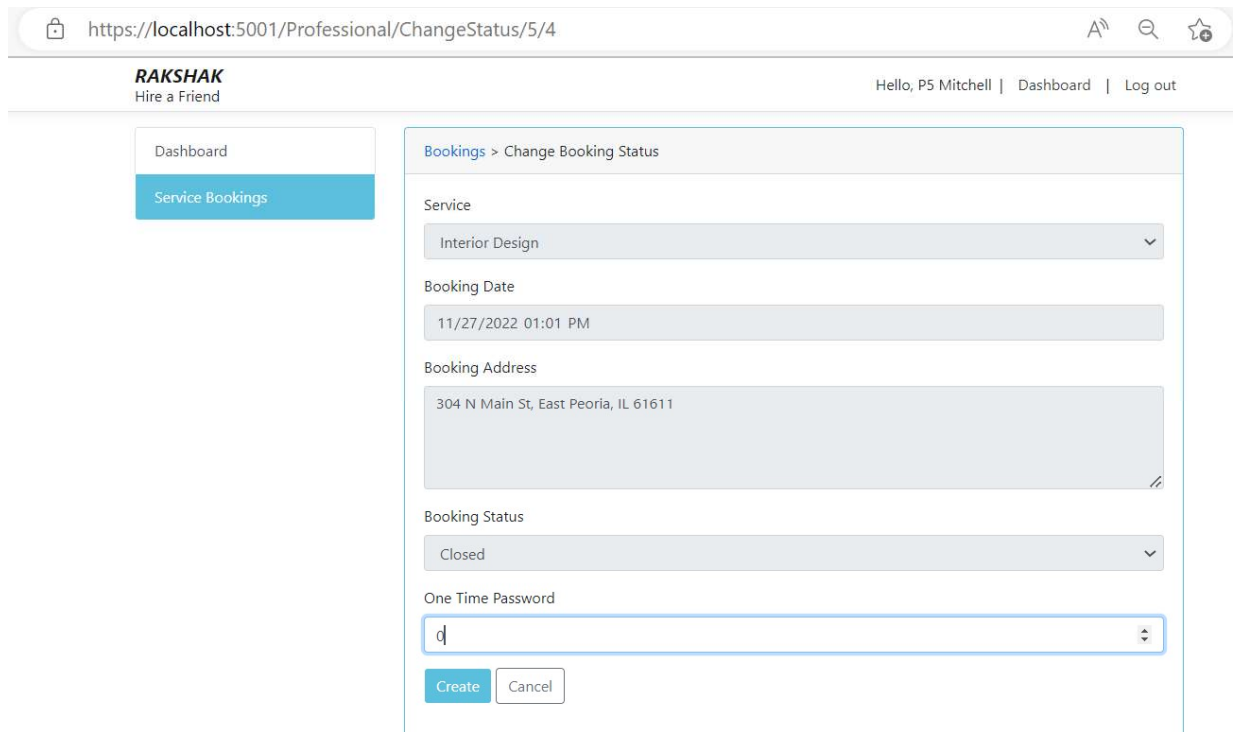


Figure 2.2.17 - Professional Closing the Assigned Booking ID:

- “Admin” facilitated with the capability to filter records based on different parameters as illustrated in Figure 2.2.18.

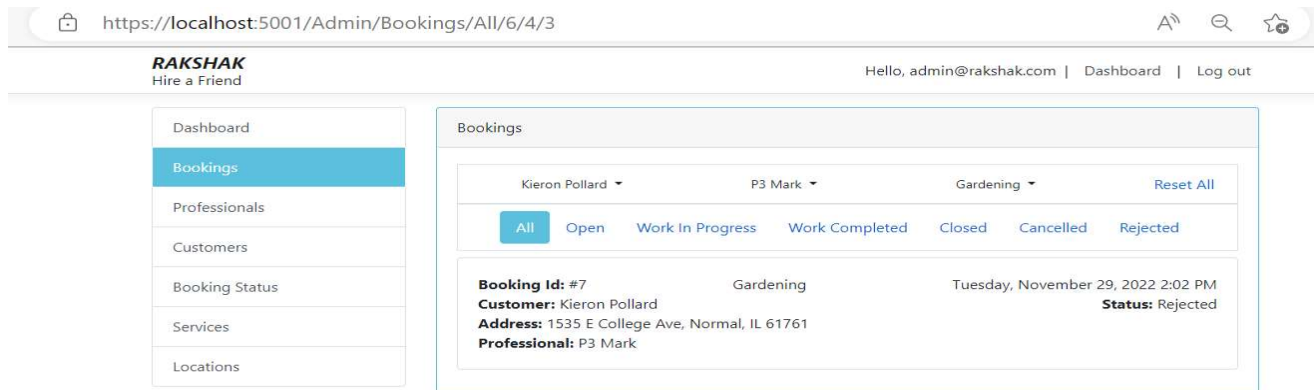


Figure 2.2.18 - Admin Filter view:

2.3 Interactions with other Applications

- We do NOT have any third party or external application interaction with our application “Rakshak”.
- Internally, we have developed different interfaces for the stakeholders (Admin, Customer, Professional) to interact & control the workflow status of each customer-initiated service request.
- Users with RoleID ‘2’ (Professionals) & ‘3’ (Customers) should exist in the dbo.Users table to create a booking service and to exercise the different types of workflow status.
- In-addition admin user should have added applicable service locations along with service types.
- Customer should provide the OTP to professional for technically close the Booking ID.

2.4 Capabilities

- Entity framework helps facilitate to configure the required database schema.
- Figure 2.4.1 would help explain the co-relation which this framework facilitates in defining the schema which acts as a base dependency to perform (create/update/retrieve) operations.
- DB context file illustrated in Figure 2.4.2 facilitates with the required getters & setter methods for data access from database tables.

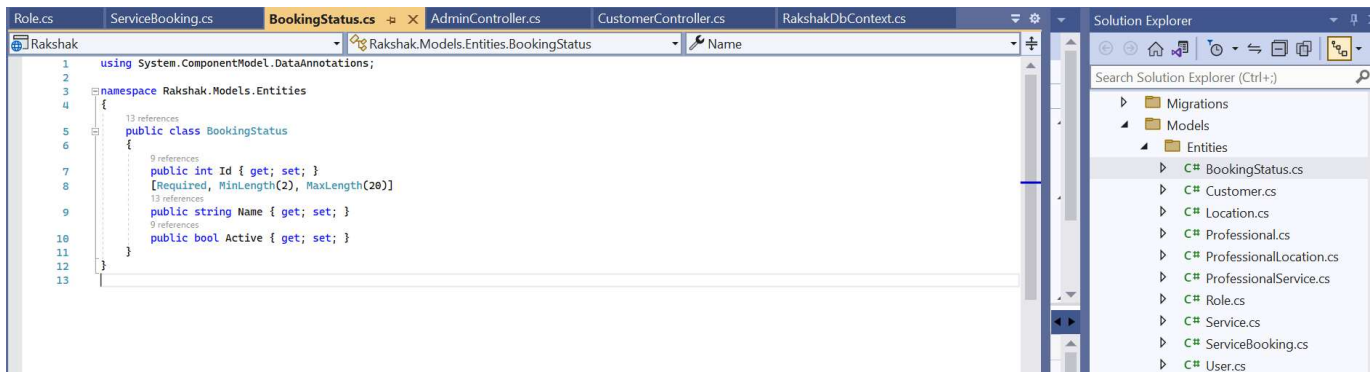


Figure 2.4.1 - Defining Database Table Schema for Booking Status:

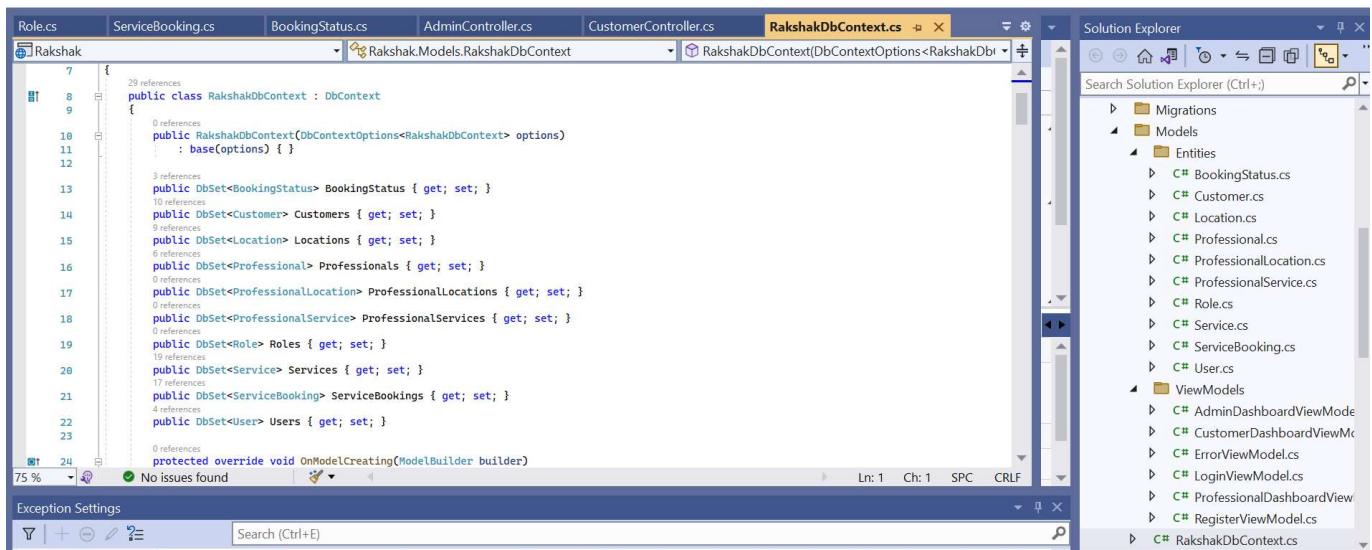


Figure 2.4.2 - DB Context with Getter & Setter Methods for Booking Status:

2.5 Risk Assessment and Management

- **Resourcing:** As a team, we should be able to back up ourselves in-terms of work sharing & collective ownership. Especially when a resource becomes unavailable (sick or emergency), other participants in the team should be able to balance the gap so that it doesn't impact the delivery schedule.
- **Re-work Effort:** Its a generic risk which we have tried mitigating as a team since from the project kick-off. We ensured that we have a proper plan in-place and regular collaboration exists between the participants of this project to avoid possible rework effort.
- **Testing & Bug Fixes:** Regular weekly defect triage calls and focus on the quality of deliverables could help facilitate & resolve showstopper bugs so that our team doesn't face a prolonged blocker during development phase.

3 Project Requirements

3.1 Identification of Requirements

The baselined requirements for Rakshak application are explained through Table 3.1.1.

REQUIREMENT ID	REQUIREMENT DESCRIPTION
RK_SP2022-1 Customer-Capability-0100	Capability to create a user with role as "Customer"
RK_SP2022-1 Customer-Capability-0200	Capability to facilitate user authorization & allow logon to Rakshak portal for only registered customers.
RK_SP2022-1 Customer-Capability-0300	Capability of "Customer" to "Create a New Service Request"
RK_SP2022-1 Customer-Capability-0400	Capability of "Customer" to "Cancel the Created Service Request".
RK_SP2022-1 Customer-Capability-0500	Capability of "Customer" to view the Dashboard of Bookings made.
RK_SP2022-1 Customer-Capability-0600	Capability of "Customer" to update his profile address.
RK_SP2022-1 Professionals-Capability-0700	Capability to create a user with role as "Professional"
RK_SP2022-1 Professionals-Capability-0800	Capability to facilitate user authorization & allow logon to Rakshak portal for only registered Professionals.
RK_SP2022-1 Professionals-Capability-0900	Capability of "Professional" to "Reject the Assigned Service Request".
RK_SP2022-1 Professionals-Capability-1000	Capability of "Professional" to view the Dashboard of Assigned Bookings.
RK_SP2022-1 Professionals-Capability-1100	Capability of "Professional" to update his profile address.
RK_SP2022-1 Admin-Capability-1200	Capability of Admin to Add/Edit Service Locations
RK_SP2022-1 Admin-Capability-1300	Capability of Admin to Add/Edit list of services
RK_SP2022-1 Admin-Capability-1400	Capability of Admin" to view the Dashboard of Assigned Bookings.
RK_SP2022-1 Admin-Capability-1500	Capability of Admin to view list of Professionals & their respective workload.
RK_SP2022-1 Admin-Capability-1600	Capability of Admin to view list of Customers & their respective Service Bookings with Status
RK_SP2022-1 Admin-Capability-1700	Capability of Admin to Assign Professional to Service Bookings
RK_SP2022-1 Admin-Capability-1800	Capability of "Admin" to "Cancel the Created Service Booking Request".
RK_SP2022-1 Admin-Capability-1900	Capability for Admin to filter records in Bookings UI

Table 3.1.1 - Requirements with Description

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

- Operational aspect of Rakshak portal would be to facilitate day-to-day business operations by ensuring that customers get their requested services and complete transparency about their service request status. System availability would be one of the key parameters while we think of Rakshak on a large-scale platform.
- The core operational performance relies on the admin interface, especially the Bookings interface where the admin could play around through the facilitated filters and assign suitable service professional to the customer-initiated requests.
- Administration of Rakshak involves collective responsibility of Admin & Professionals. More importantly, while we think of its implementation on a larger scale, core expectation of administration would be admin access to portal using common password, work plan, ensuring the database is up & running a sanity check on the Users/Booking database tables to ensure data integrity remains intact.
- Maintenance aspect of Rakshak involves consistently reviewing the service locations and list of offered services as organizations may scale over a period and depending upon business needs, administrator should be able to keep this information updated. Handling further upgrades to framework or libraries should be planned clearly with proactive communication addressed to service professionals and customers through scheduled outage.

- In case of future enhancements, the new features added or integration with external applications should be planned with proper data back up as it holds the key information for the execution of everyday business. Also in such cases, the cross dependency between different teams & their responsibilities should be well communicated & planned before implementation.

3.3 Security and Fraud Prevention

- Access to Rakshak portal is ONLY facilitated to users on successful Authorization.
- Users' password choice should be a combination of (one uppercase, lowercase, special character and numeric) which would ensure each user has strong password which is not that easy to judge and assures safety for users from external threats.
- Service Requests can be closed by Professionals in Rakshak portal ONLY if the OTP provided matches with the system expectation.
- ONLY "Admin" user has the capability to assign service professional to service requests.
- ONLY the assigned "Professional" user will have the capability to move forward with workflow status for ONLY the Assigned service request. ((i.e.,) Changing status from "Open" till "Work Completed" & "Closed")

3.4 Release and Transition Plan

- Communicate the deployment plan to customer stake holders well in advance.
- Deploy the published Rakshak application to a folder in the hosting server.
- Setup a process manager that starts the application when requests arrive and restarts the app after it crashes.
- Perform smoke test in production environment post deployment.
- End to End workflow transition should be facilitated to customer stake holders prior to the release schedule.
- Detailed project documentation on the workflow with screenshots and flows should be provided to the customer stakeholders.
- Production support schedule & SLA agreement should be outlined and communicated to Customer stakeholders prior to implementation.
- Data back up and system recovery plan should be outlined clearly in case of unforeseen incidents & to support business continuity.

4 Project Design Description

- Model-View-Controller (MVC) architectural pattern separates an application into three main groups of components.
- Model component in an application represents the state of the application and any operations that should be performed by it. Business logic would be encapsulated in the model along with any implementation logic for persisting the state of the application.
- Views facilitates presenting content through the user interface. It utilizes the Razor view engine to embed .NET code in HTML markup.
- Controller handles and responds to user input and interaction. Controller selects which model types to consider, and which view to be rendered.

- **Convention-based routing** enables to globally define the URL formats that your application accepts and how each of those format's maps to a specific action method on a given controller.

Routing in Rakshak application is being configured by the below lines of statement in DbContext.CS file.

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}")
    }
}
```

- **Model validation** is performed by decorating model object with data annotation validation attributes. The validation attributes are validated on the client side before values are posted to the server, as well as on the server before the controller action is called.

Below is a sample implementation to demonstrate **Model validation in Rakshak application**.

```
public class ServiceBooking
{
    public int Id { get; set; }
    public int CustomerId { get; set; }
    [Required, Display(Name = "Service")]
    public int ServiceId { get; set; }
    [Display(Name = "Professional")]
    public int? ProfessionalId { get; set; }
    [Display(Name = "Booking Date")]
    public DateTime BookingOn { get; set; }
    [Required, Display(Name = "Booking Address")]
    public string BookingAddress { get; set; }
    [Display(Name = "Booking Status")]
    public int BookingStatusId { get; set; }
    public string Remarks { get; set; }
    public int OTP { get; set; }
    public string Rating { get; set; }
    public DateTime DateCreated { get; set; }

    public Professional Professional { get; set; }
    public Customer Customer { get; set; }
    public Service Service { get; set; }
    public BookingStatus BookingStatus { get; set; }
}
}
```

Subsequent controller action is demonstrated in the below lines of statement:

```
public IActionResult NewBooking(ServiceBooking booking)
{
    ViewBag.Services = _context.Services.Where(x => x.Active)
        .Select(x => new SelectListItem { Value = x.Id.ToString(), Text = x.Name }).ToList()

    if (ModelState.IsValid)
    {
        var userId = User.Claims.First(x => x.Type == ClaimTypes.NameIdentifier).Value
            var customerId = _context.Customers.Where(x => x.UserId == Convert.ToInt32(userId)).Select(x =>
x.Id).FirstOrDefault()
        booking.CustomerId = customerId
        booking.BookingStatusId = 1
        booking.DateCreated = DateTime.Now
        booking.OTP = GenerateOTP()

        _context.ServiceBookings.Add(booking)
        _context.SaveChanges()

        return RedirectToAction("ServiceBookings")
    }
    return View(booking)
}
```

- **Utilizing entity framework** for Database schema creation for Rakshak as the below **Figure 4.0.1** would serve the purpose of illustration.

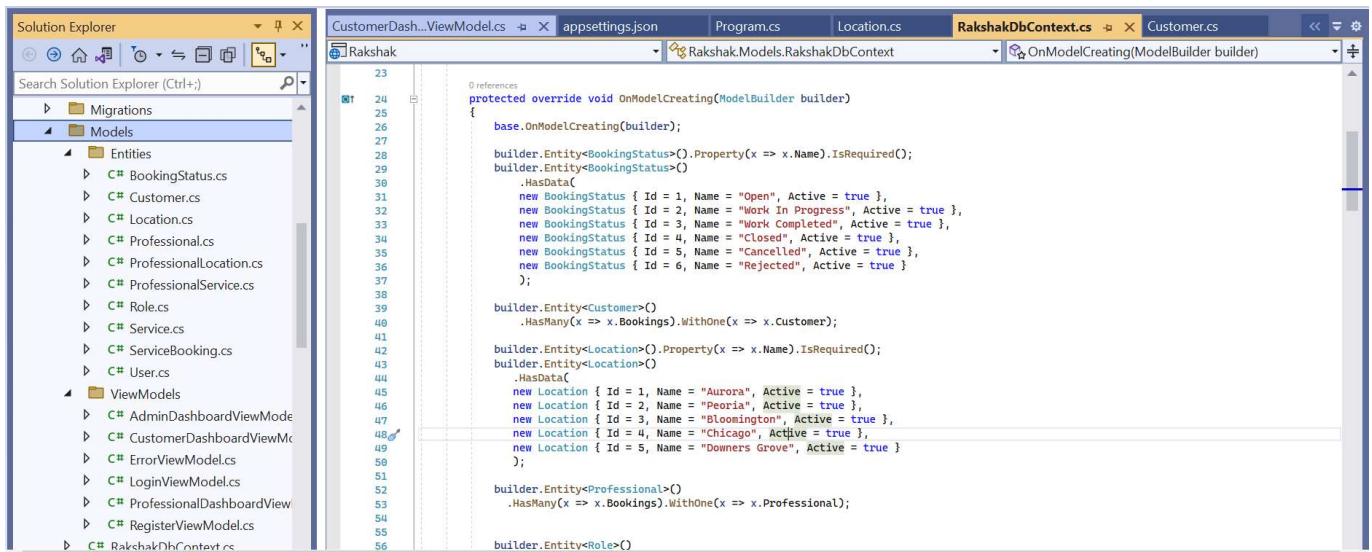


Figure 4.0.1 - Utilizing Entity Framework

5 Internal/external Interface Impacts and Specification

- Rakshak solution has been interfaced with Logger factory (Figure 5.0.1).method which could help facilitate providing logger information based on the user operations. (.UseLoggerFactory(LoggerFactory.Create(builder => builder.AddConsole()))

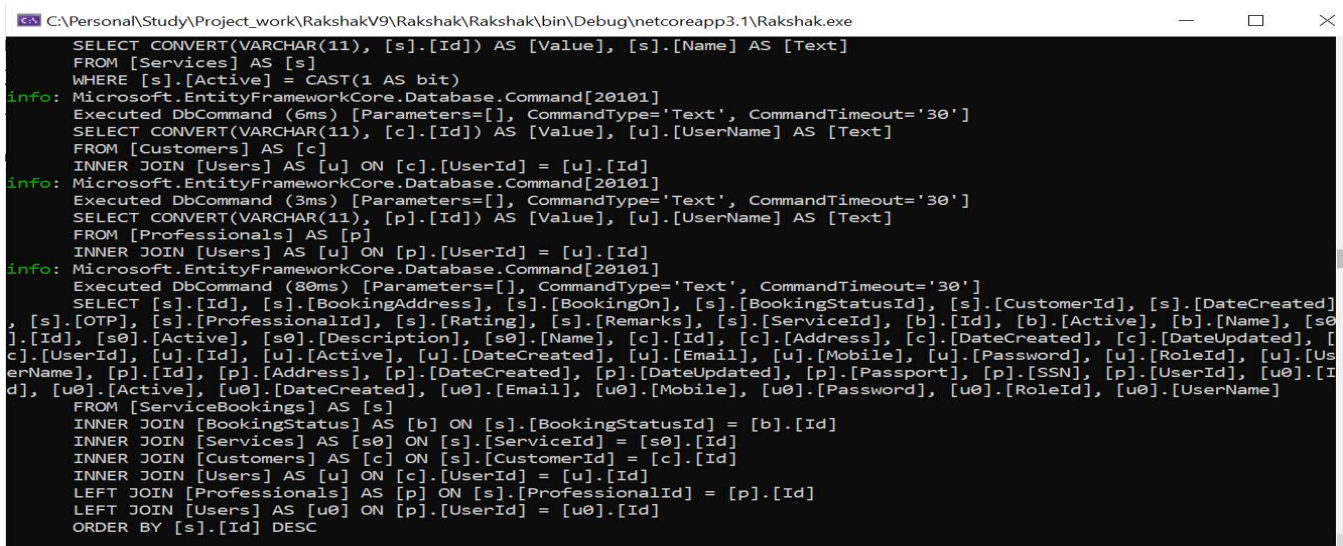


Figure 5.0.1 - Interface with Logger Factory

- Persistent data explained in Figure 5.0.2 is configured in the below database tables as these configurations may not change frequently (Ex: dbo.Roles)
NOTE*: New Service locations and list of Services could be added by administrator through Rakshak user interface based on requirements.

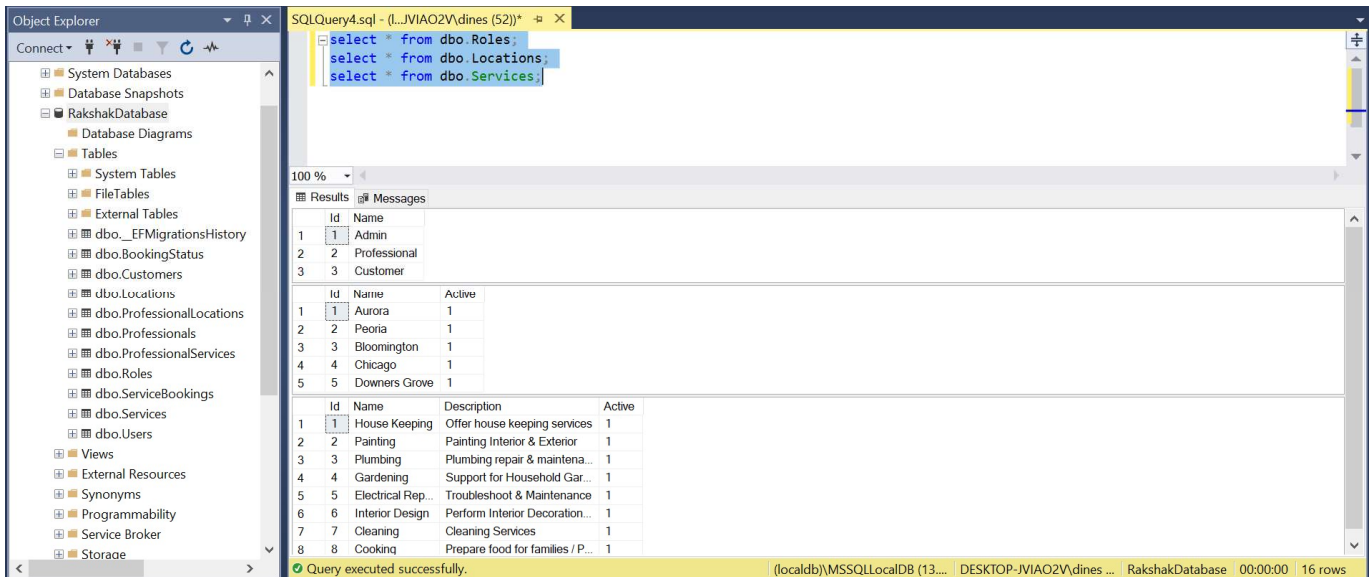


Figure 5.0.2 - Service Locations & List of Services Managed by Administrator

6 Design Units Impacts

The below stated UI design units are inter-dependent and demonstrate the core feature of Rakshak web interface. Model component in an application represents the state of the application and any operations that should be performed by it. Business logic would be encapsulated in the model along with any implementation logic for persisting the state of the application. Views facilitates presenting content through the user interface. It utilizes the Razor view engine to embed .NET code in HTML markup. Controller handles and responds to user input and interaction. Controller selects which model types to consider, and which view to be rendered.

- Account: Login & Register
- Admin: Assign Professional, Booking Status
- Customer: Create New /Edit Services Bookings, Update Profile
- Professionals: Change Status

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

Web interface unit in Rakshak represents the overall end to end operational workflow and acts as an enabler between the key stakeholders (Admin, Customer and Professional). Example: Booking UI interface (Figure 6.1.1(A)) is an inter-dependent functional unit for “Admin”, and it also represents the workload for everyday operations. Admin usually plays around with this key feature on everyday basis and performs tasks like assigning professionals to service requests, adding new locations or services. Bookings user interface has filters which could help “Admin” to fetch records based on Customers or Professionals or Particular Service type. Also, Add/Edit Service List & Locations UI interface (Figure 6.1.1(B)) is another example for inter-dependent functional unit. Though it could only be controlled by “Admin”, the other workflow interfaces are dependent on this functional unit.

https://localhost:5001/Admin/Bookings/All/0/0/0

RAKSHAK
Hire a Friend

Hello, admin@rakshak.com | Dashboard | Log out

- Dashboard
- Bookings**
- Professionals
- Customers
- Booking Status
- Services
- Locations

Bookings

All Customers All Professionals All Services [Reset All](#)

All Open Work In Progress Work Completed Closed Cancelled Rejected

Booking Id: #8 **Service:** Cleaning **Date:** Sunday, November 27, 2022 12:00 AM
Customer: Daniel Martyn **Status:** Open
Address: 304 N Main St, East Peoria, IL 61611
Professional: NA [Assign Professional](#) | [Cancel Booking](#)

Booking Id: #7 **Service:** Gardening **Date:** Tuesday, November 29, 2022 2:02 PM
Customer: Kieron Pollard **Status:** Rejected
Address: 1535 E College Ave, Normal, IL 61761
Professional: P3 Mark

Booking Id: #6 **Service:** Cleaning **Date:** Monday, November 28, 2022 5:05 PM
Customer: Daniel Martyn **Status:** Cancelled
Address: 304 N Main St, East Peoria, IL 61611
Professional: NA
Remarks: Change of Plan. Cancelling this request

Booking Id: #5 **Service:** Interior Design **Date:** Sunday, November 27, 2022 1:01 PM
Customer: Daniel Martyn **Status:** Work Completed
Address: 304 N Main St, East Peoria, IL 61611
Professional: P5 Mitchell

Figure 6.1.1(A) - Admin Bookings User Interface with Filters:

https://localhost:5001/Services

RAKSHAK
Hire a Friend

Hello, admin@rakshak.com | Dashboard | Log out

- Dashboard
- Bookings
- Professionals
- Customers
- Booking Status
- Services**
- Locations

Services [+ Create New](#)

#	Name	Description	Active	Actions
1	House Keeping	Offer house keeping services	Active	Edit Bookings
2	Painting	Painting Interior & Exterior	Active	Edit Bookings
3	Plumbing	Plumbing repair & maintenance services	Active	Edit Bookings
4	Gardening	Support for Household Garden Maintenance	Active	Edit Bookings
5	Electrical Repairs	Troubleshoot & Maintenance	Active	Edit Bookings
6	Interior Design	Perform Interior Decorations for Special Occasions	Active	Edit Bookings
7	Cleaning	Cleaning Services	Active	Edit Bookings
8	Cooking	Prepare food for families / Party & Special Occasions	Active	Edit Bookings

Figure 6.1.1(B) - Add/Edit Service List & Locations

6.1.2 Impacts

In case of any enhancements or edits to these web interfaces (views), the corresponding model and controller action components should be changed in parallel for the entire functional unit to perform per the requirements. In other words, model view and controller components are inter-dependent and work together to ensure that the desired functionality of the design unit works per the expectations.

6.1.3 Requirements

- Capability of Admin to Add/Edit Service Locations
- Capability of Admin to Add/Edit list of services
- Capability of Admin" to view the Dashboard of Assigned Bookings.
- Capability of Admin to view list of Professionals & their respective workload.
- Capability of Admin to view list of Customers & their respective Service Bookings with Status.
- Capability of Admin to Assign Professional to Service Bookings.
- Capability of "Admin to "Cancel the Created Service Booking Request".
- Capability for Admin to filter records in Bookings UI

6.2 Functional Area A/Design Unit A

6.2.1 Functional Overview

Database unit is another inter-dependent functional unit (Figure 6.2.1(A)) which helps manage the data required for the application web interfaces. All the data pertaining to different roles (admin, customers, professionals) are being maintained in appropriate tables within Database. (Example: dbo.Users, dbo.Customers, dbo.Professionals). All the customer initiated booking service requests are being managed in dbo.ServiceBookings database table

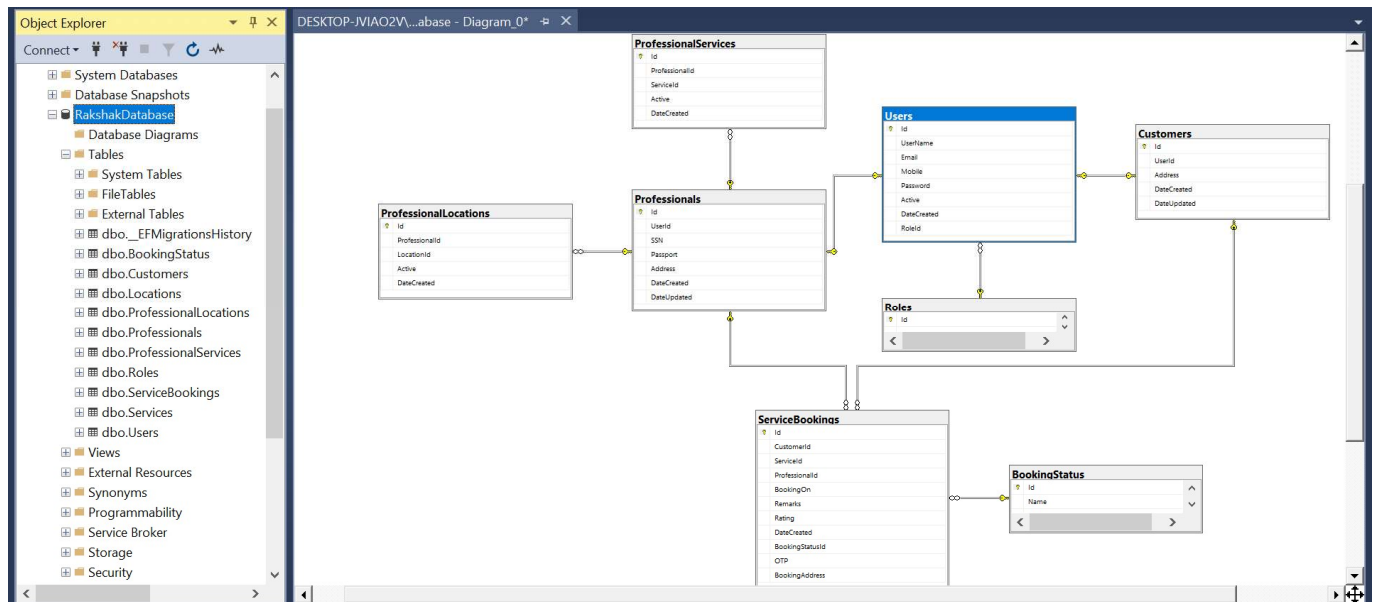


Figure 6.2.1 (A) - Database Entity Relationship

S

6.2.2 Impacts

In case of future enhancements or any edits to existing database table schema, we need to ensure that the data integrity remains the same and all the existing operations performed by the web interface units perform the same way without causing any issues while creating new users (Customers, Professionals) or while moving forward with the workflow status as "Service Professional".

6.2.3 Requirements

- Registered customers profile data should be updated/retrieved from dbo.Users & dbo.Customers database table.
- Registered professionals profile data should be updated/retrieved from dbo.Users & dbo.Professionals database table.
- Customer initiated service bookings data should be managed through dbo.ServiceBookings database table.
- Service Locations data should be updated/retrieved from dbo.Locations database table.
- List of Services should be updated/retrieved from dbo.Services database table.

7 Acknowledgements

We would like to thank our professor Dae Wook Kim for assisting us & providing us with the required direction for this project work. Our weekly progress demo report and consistent connect with professor really helped us receive constructive feedbacks which in turn helped to improve our knowledge and our overall product quality. We would also like to thank each of our team participants in Group#3 project who collaboratively worked together in order to meet the project goals in the scheduled timeline. We would also like to extend our thanks to “Department of Computer Science” for giving us this opportunity to research, explore, collaboratively design & develop and for facilitating us to experience a complete end to end project work as part of this course work.

8 References

- [1] **Overview of entity Framework Core - EF Core. Overview of Entity Framework Core - EF Core** | Microsoft Learn, from <https://docs.microsoft.com/en-us/ef/core/>
- [2] **Overview of ASP.NET core MVC.** Microsoft Learn, from [What's new in ASP.NET Core 3.1 | Microsoft Learn](https://docs.microsoft.com/en-us/aspnet/core/mvc/what-is-new-in-asp-net-core-3-1?WT.mc_id=dotnet-351-javascript)
- [3] **IAMTimCorey. (2020, January 13). Entity framework best practices** - YouTube, from <https://www.youtube.com/watch?v=qkJ9keBmQWo>
- [4] **Mark Otto, J. T. Overview.** Bootstrap v4.6, from <https://getbootstrap.com/docs/4.6/layout/overview/>
- [5] **Top 10 Home Services Applications, Codiant** from <https://www.codiant.com/blog/top-10-on-demand-home-service-apps>
- [6] **Top Reasons Why On-Demand Home Services App Have Bright future, (2021, June 28)** Cerebruminfotech, from <https://www.cerebruminfotech.com/blog/top-reasons-why-on-demand-home-services-apps-have-a-bright-future>