Attachment IV: Functional Design Document

Muhammad Tariq Arif Hussain
FONTYS UNIVERSITY OF APPLIED SCIENCES |

TITLE PAGE

HBO-ICT: English Stream

Data Project Manager:	
Family name, initials:	**Hussain, M.T.A.H.**
Student number:	**2222716**
project period: (from – till)	**August 27, 2018 – February 1, 2019**
Data Steering Committee:	
Company:	**OCS-Consulting B.V.**
Department:	**Consulting Management**
Address:	**Ruwekampweg 2G** **5222 AT 's-Hertogenbosch** **The Netherlands**
Product Owner:	
Family name, initials:	**Ebben, R.E.**
Company Tutor:	
Family name, initials:	**Zanella, R. Z.**
Position:	**SAS-Consultant**
Document:	
Title:	**Functional Design Document**
Version:	**4**
Date:	**November 19th, 2018**

Document history

Revisions

Version	Status	Date	Changes
1	Concept	October 8	Added Management Summary, Introduction, Owner's Expectation, Database Model and Web-Application Wireframe.
2	Draft	October 15	Changed Database Model and Web-Application Wireframe. Added ERD Association of Wireframe, Functional Requirements Mapping, Tools Used.
3	Draft	October 17	Changed Database Model, Web-Application Wireframe.
4	Final	November 16	Added tables in Database Model and changed SCREEN 14 of Web-Application Wireframe. Changed 'Client' to 'Owner'.

Approval

This document requires following approvals:

Version	Date approval	Name	Function	Initials
3	October 17th,2018	Rocco Zanella	Company Tutor	R. Z.

Distribution

This document is distributed to:

Version	Date distribution	Name	Function
1	October 12, 2018	Rocco Zanella	Company Tutor
2	October 16, 2018	Rocco Zanella	Company Tutor
3	October 17, 2018	Rocco Zanella	Company Tutor

Management Summary

Document Objective

The aim of this document is to

- define Product Owner's (herein after Owner) Expectation
- define Database Model
- state the steps taken to create Database Model
- present Database Model
- define Web-Application Wireframe
- state the steps taken to create Web-Application Wireframe
- present Web-Application Wireframe

in **Project: Web-Application for managing clientele information in OCS-Consulting[1],** referred hereafter as Project.

Motivation

The functional design of Owner's Expectation in its entirety, is made to provide a conceptual foundation for its development into software. Further, the functional requirement of a database to meet Owner's Expectation is explained in this document.

Global Approach

The approach is divided into two phases. In the first phase, the physical Entity Relationship Diagram of Database Model is created, and the steps taken for its creation is documented.

In the second phase, Owner's Expectation defined in **MoSCoW Requirements Prioritization[2]** is conceptualized into a functional design by user interface diagrams.

[1] This project was initiated in OCS Consulting B.V. to develop a web-application for storing and managing information about their clients in an RDBMS and leveraging its properties for generating contracts.

[2] The MoSCow Requirements Prioritization is a component of the project which identifies and prioritizes the requirements from the application into "Must Have", "Should Have", "Could Have", "Will Not Have" categories.

Contents

1. Introduction

This document has been written to define the functional requirements of a database and the design of a web-application to meet Owner's Expectation.

The **Functional Design Document** covers the following fundamental aspects of Project:

- Definitions and method of making Database Model and Web-Application Wireframe.
- Definition of Owner's Expectation.
- Functional development of Owner's Expectation.

The document is used to:

- Ensure that Project covers functional requirements for developing the web-application to Owner's Expectation.
- Provide Project Members a conceptual standpoint for **Alpha Functional Model** and **Beta Functional Model**[3]

Further, certain assumptions are made to deliver a technically acceptable design with Functional Design Document. These assumptions are the technical constraints applied to accommodate the functionality expected by Owner from a software product. Following are the assumptions made:

- **Owner's expectation is met with a form of software product, i.e. a web application running on a local server.**
- **Its functions include data entry, storage, removal and modification to accommodate Process, i.e. the current process with which Owner manages their own tasks/roles.**
- **The relational database at its back end is represented by Database Model.**
- **The presentation layer or its front end is represented by Web-Application Wireframe**.

[3] The two prototypes that were tested and delivered. The later was developed further to meet more of the requirements prescribed.

2. Owner's Expectation

The Owner's expectation from the Project in terms of requirements are studied and presented in this chapter. Further, a clear understanding of the Owner's expectation helps to determine whether it is a functional or a technical requirement. This in turn helps prioritise the requirement(s) for Alpha Functional Model and Beta Functional Model within the scope of Project. These expectations are presented below:

Exp. ID	**DESCRIPTION**
1.	To manage clientele information of OCS-Consulting B.V., replacing Excel spreadsheet overviews of ongoing projects.
2.	To generate Contracts for OCS-Consulting replacing MS-Word.
3.	To have a single page for data entry and submission for generating Contracts.
4.	To automatically increment Project Numbers and show it in the generated overview when new projects are added.
5.	To implement both ENGLISH and DUTCH Contract templates.
6.	To use information from other tables as a choice, rather than input the same information into text-fields each time it is needed.
7.	To share the status of a project as ongoing or finished.
8.	To highlight projects nearing their end-date.

3. Database Model

In this chapter, the Database Model is defined and presented. It is the physical Entity Relationship Diagram (ERD) of a web-application database designed to meet Owner's expectation. The web-application stores information and this can be achieved with a relational database. The technical requirement of a physical Entity Relationship Diagram is to help with designing a user interface or a wire frame for the web-application.

Further, the creation of Database Model is achieved by strategic arrangement of its tables and by using a method of planning based on conception. The utility of describing this method is its reference when planning STAGE 3 of Global Planning. The steps taken successively are as follows:

STEP 1: Collect all resources used by Owner for managing Company clientele information, i.e. Assignment Confirmation Form(s) hereby referred to as Contracts and Excel spreadsheet overviews:

- Overview of current contracts – assignments.xlsx
- Project number overview.xlsx

STEP 2: Select the stream of Contract templates to work first, between ENGLISH and DUTCH.

STEP 3: List all fields within selected Contract templates, and Excel spreadsheet overviews.
Fields are the columns for tables to be arranged in the database.

STEP 4: Remove duplicates.
In this step, duplicates are available in name and in meaning. Hence, duplicates are removed in both name and meaning. This is the database in its First Normal Form (1NF).

STEP 5: Consider five entities for arranging the fields, namely: Manager, Projects, Contracts, Clients, Consultants.
This is a strategic arrangement of tables. Here, entities are not arranged based on partial dependency, but on functional dependency. An educated estimation (confirmed by Company Tutor) is made in this arrangement to factor out the time needed for removing any partial dependency and transitive dependency to achieve Second Normal Form and Third Normal Form respectively.

STEP 6: Arrange fields into entities as shown:

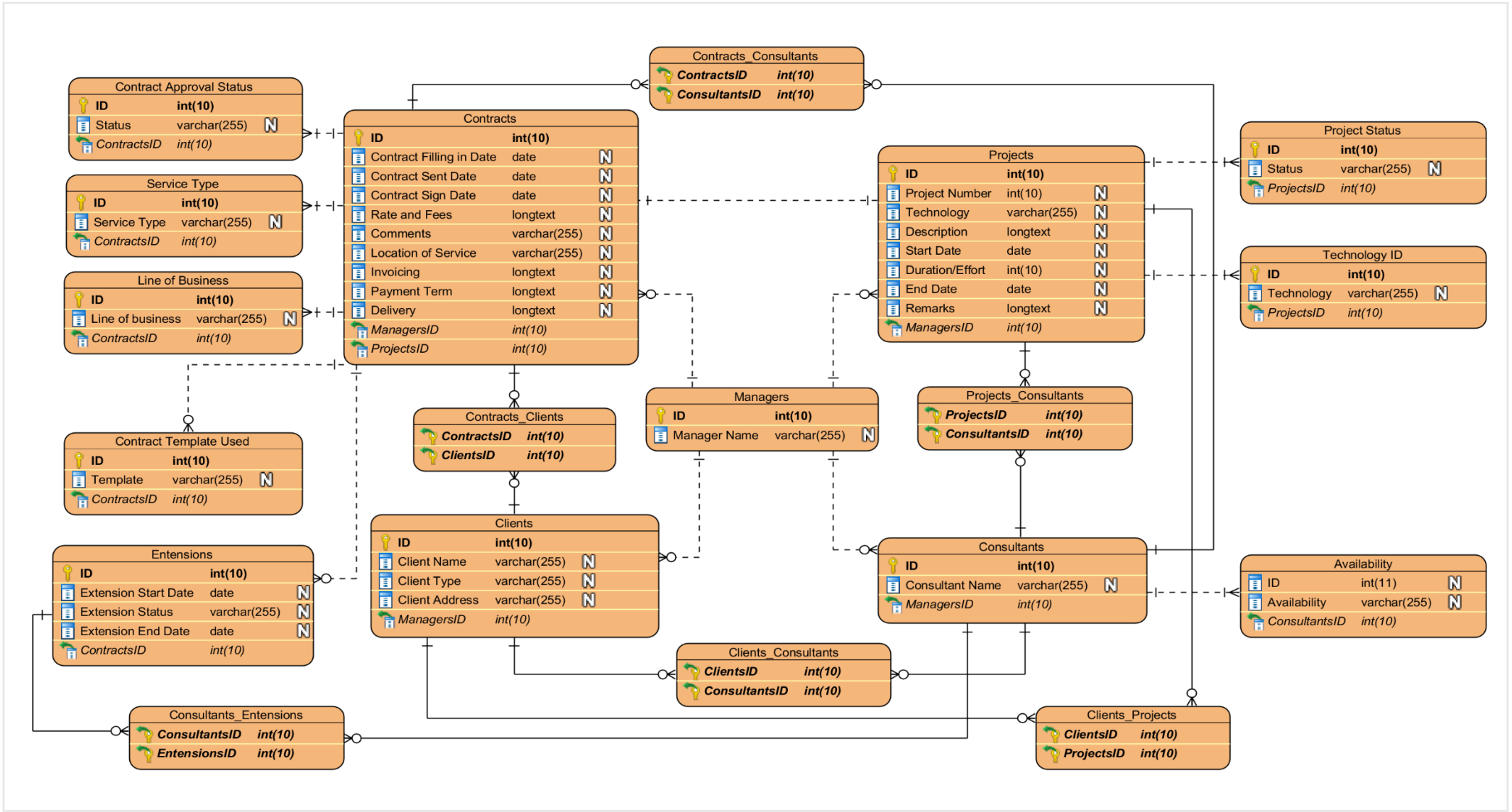
Manager	Projects	Contracts	Clients	Consultants
• Manager Name	• Project Number • Technology • Description • Start Date • Duration • End Date • Remarks	• Contract Filling in Date • Contract Sent Date • Contract Sign Date • Rate and Fees • Comments • Location of Service • Invoicing • Payment Term • Delivery • Line of Business	• Client Name • Service Type • Client Address	• Consultant Name • Availability

STEP 7: Define the unique relationships between entities as shown:

Managers table	Projects table	Contracts table	Clients table	Consultants table
• Manager with Projects (One to Many) • Manager with Contracts (One to Many) • Manager with Clients (One to Many) • Manager with Consultants (One to Many)	• Projects with Contracts (One to One) • Projects with Clients (Many to Many) • Projects with Consultants (Many to Many) • Projects with Project Status (One to Many) • Projects with Technology (One to Many)	• Contracts with Clients (Many to Many) • Contracts with Consultants (Many to Many) • Contracts with Extensions (One to Many) • Contracts with Line of Business (One to Many) • Contracts with Contract Template Used (One to Many) • Contracts with Service type (One to Many) • Contracts with Contract Approval Status (One to Many)	• Clients with Consultants (Many to Many)	• Consultants with Availability (One to Many)

These relationships are defined within the context of Owner's expectation of a single page for data entry and submission to create a Contract with the web-application.

STEP 8: Create a physical ERD diagram as presented below.



4. Web-Application Wireframe

In this chapter, the Web-Application Wireframe is defined and presented. It is the mock-up and the non-functional prototype of a web-application wireframe designed to meet Owner's expectation. The aim of the mock-up and the prototype is to deliver a conceptual understanding of the web-application with medium and high-fidelity representations respectively. Representative fidelity used in Web-Application Wireframe is defined below:

- Low-fidelity representation: A hand-drawn sketch of the wireframe.
- Medium-fidelity representation: A mock-up with static images of the wireframe.
- High-fidelity representation: A non-functional prototype with interactive components.

Further, the process by which the mock-up and the non-functional prototype created is described for its evaluation and reference in STAGE 3. The steps taken successively are as follows:

STEP 1: Visualize Owner's interaction with the web-application.
STEP 2: Predetermine Owner's interaction with the web-application to input clientele information, according to the physical Entity Relationship Diagram (ERD).
STEP 3: Sketch out mock-ups by hand for visual support, mapped to the predetermined Owner interaction.
STEP 4: Create mock-ups based on the sketches.

4.1 Mock-up Breakdown Structure

The Mock-up Breakdown Structure is defined to represent its twenty-seven elements with Fig.1, Fig.2, Fig.3, Fig.4, Fig.5, Fig.6 and Fig.7 as shown below. Further each representation is supplied with a table and a column (**Functional Description)**, to define its function and type i.e. interactive(clickable) and indicative.

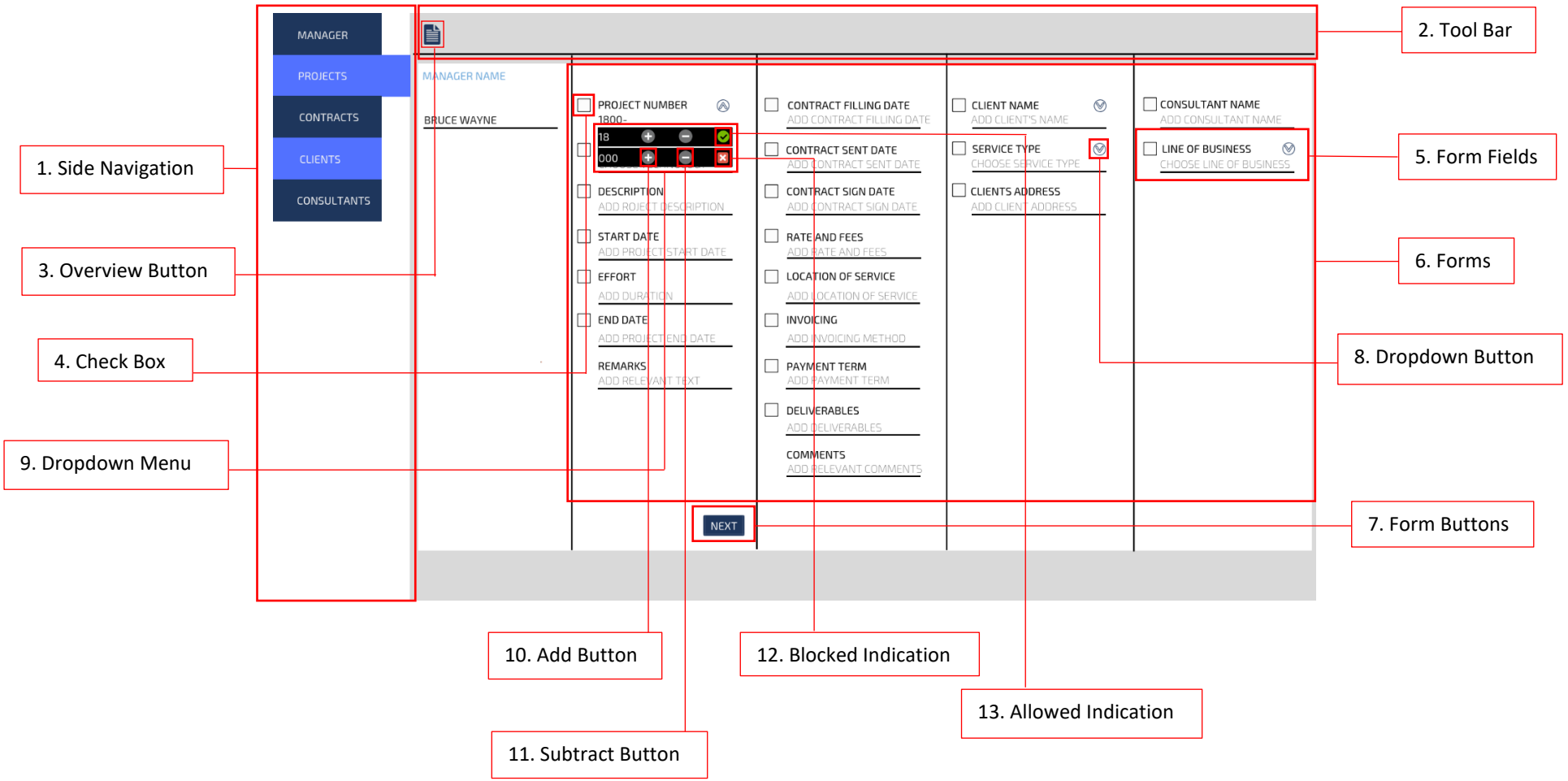

MANAGER
PROJECTS
CONTRACTS
CLIENTS
CONSULTANTS
MANAGER NAME
BRUCE WAYNE
PROJECT NUMBER
1800-
18
000
DESCRIPTION
ADD PROJECT DESCRIPTION
START DATE
ADD PROJECT START DATE
EFFORT
ADD DURATION
END DATE
ADD PROJECT END DATE
REMARKS
ADD RELEVANT TEXT
CONTRACT FILLING DATE
ADD CONTRACT FILLING DATE
CONTRACT SENT DATE
ADD CONTRACT SENT DATE
CONTRACT SIGN DATE
ADD CONTRACT SIGN DATE
RATE AND FEES
ADD RATE AND FEES
LOCATION OF SERVICE
ADD LOCATION OF SERVICE
INVOICING
ADD INVOICING METHOD
PAYMENT TERM
ADD PAYMENT TERM
DELIVERABLES
ADD DELIVERABLES
COMMENTS
ADD RELEVANT COMMENTS
CLIENT NAME
ADD CLIENT'S NAME
SERVICE TYPE
CHOOSE SERVICE TYPE
CLIENTS ADDRESS
ADD CLIENT ADDRESS
CONSULTANT NAME
ADD CONSULTANT NAME
LINE OF BUSINESS
CHOOSE LINE OF BUSINESS
NEXT
1. Side Navigation
2. Tool Bar
3. Overview Button
4. Check Box
5. Form Fields
6. Forms
7. Form Buttons
8. Dropdown Button
9. Dropdown Menu
10. Add Button
11. Subtract Button
12. Blocked Indication
13. Allowed Indication

Fig.1

ID	Mock-up Breakdown Structure- Fig.1	Functional Description
1.	Side Navigation	Side Navigation covers the five tabs namely, Manager, Projects, Contracts, Clients and Consultants. Out of the five tabs, the Contracts, Clients and the Consultants tabs are interactive mediums for their associated Overviews.
2.	Tool Bar	Tool Bar covers the ten interactive mediums namely, Overview Button, Create Button, Remove Button, Generate Button, Download Button, Upload Button, Approve Button, Unapproved Button, Previous Button and Next Button.
3.	Overview Button	Overview Button is an interactive medium to visualize Overviews.
4.	Check Box	Check Box is an interactive medium to confirm selection of Form Field(s), Overviews and certain Dropdown Menu option(s).
5.	Form Fields	Form Fields are interactive mediums to insert text and numbers relevant to Process. They are associated with the fields of Database Model- STAGE 2.
6.	Forms	Forms covers the four forms of clientele data entry. Each form has a pre-defined set of Form Fields.
7.	Form Buttons	Form Buttons are an interactive medium to switch between the four forms. There are three Form Buttons. • NEXT • BACK • SUBMIT
8.	Dropdown Button	Dropdown Button is an interactive medium to select from a set of data entry inputs defined for certain Form Fields.
9.	Dropdown Menu	Dropdown Menu covers the inputs for selection from a pre-defined set for certain Form Fields.
10.	Add Button	Add Button is an interactive medium of Dropdown Menu where selected numbers can be incremented by one with each click.
11.	Subtract Button	Subtract Button is an interactive medium of Dropdown Menu where selected numbers can be reduced by one with each click.
12.	Blocked Indication	Blocked Indication is an indicative medium that represents the rejection of certain data entries within Dropdown Menu.
13.	Allowed Indication	Allowed Indication is an indicative medium that represents the acceptance of certain data entries within Dropdown Menu.

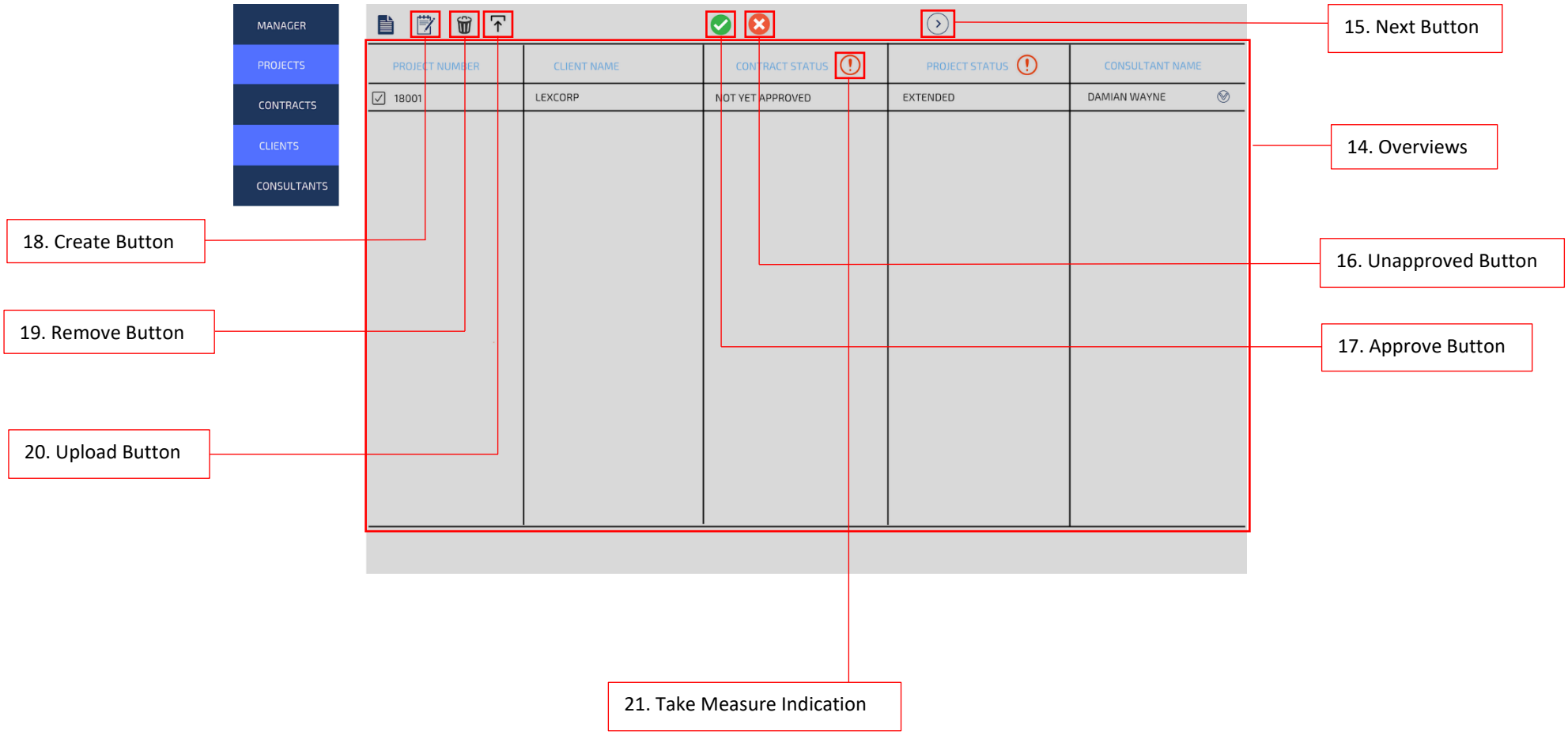

MANAGER
PROJECTS
CONTRACTS
CLIENTS
CONSULTANTS
PROJECT NUMBER
CLIENT NAME
CONTRACT STATUS
PROJECT STATUS
CONSULTANT NAME
18001
LEXCORP
NOT YET APPROVED
EXTENDED
DAMIAN WAYNE
15. Next Button
14. Overviews
16. Unapproved Button
17. Approve Button
18. Create Button
19. Remove Button
20. Upload Button
21. Take Measure Indication

Fig.2

ID	Mock-up Breakdown Structure- Fig.2	Functional Description
14.	Overviews	Overviews provides a tabular presentation of data submitted to the application. They are associated with the entities of Database Model.
15.	Next Button	Next Button is an interactive medium to change Overviews.
16.	Unapproved Button	Unapproved Button is an interactive medium to change Overviews.
17.	Approve Button	Approve Button is an interactive medium to change Overviews.
18.	Create Button	Create Button is an interactive medium to engage Forms.
19.	Remove Button	Remove Button is an interactive medium to change Overviews.
20.	Upload Button	Upload Button is an interactive medium to change Overviews.
21.	Take Measure Indication	Take Measure Indication is an indicative medium that represents the call of action due to changes in Overviews.

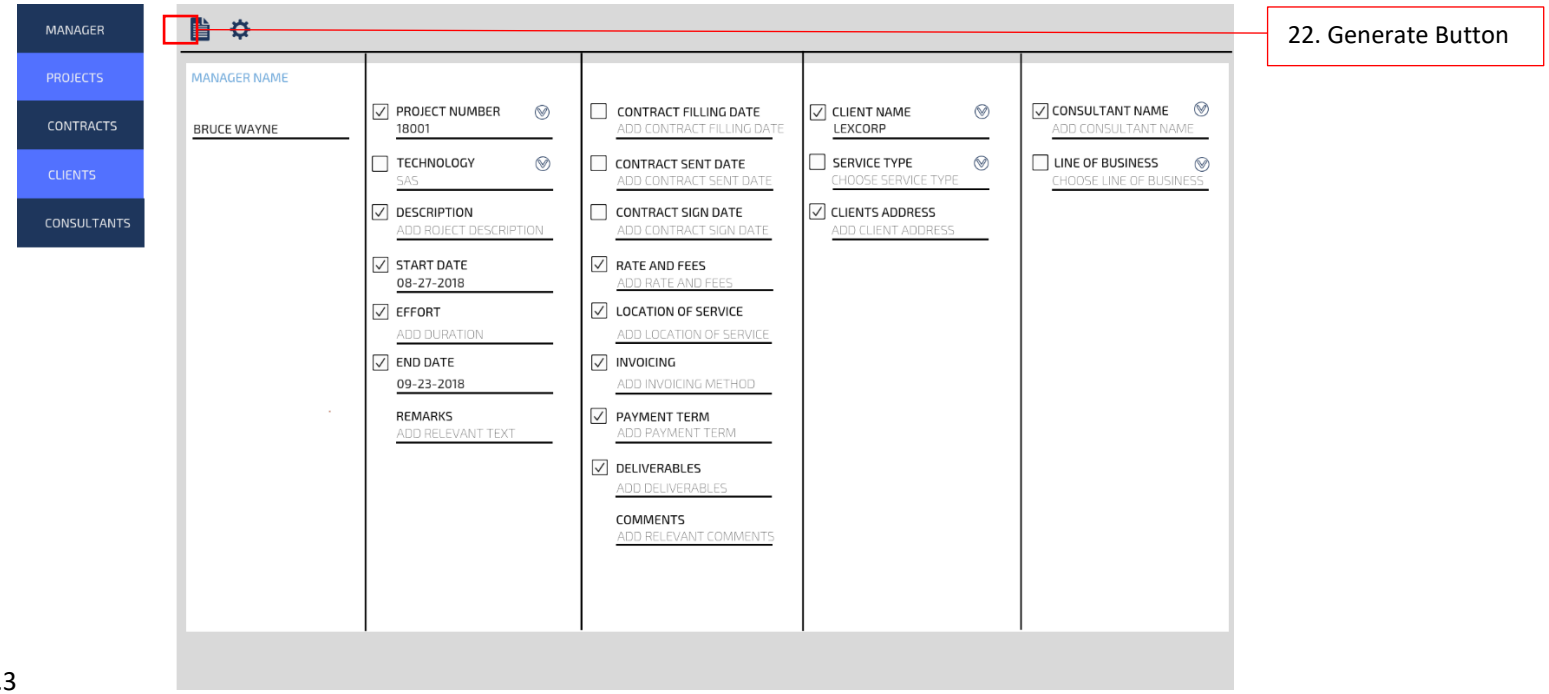


Fig.3

ID	Mock-up Breakdown Structure- Fig.3	Functional Description
22.	Generate Button	Generate Button is an interactive medium to save Form Fields selected by filling their respective Check Box for arrangement.

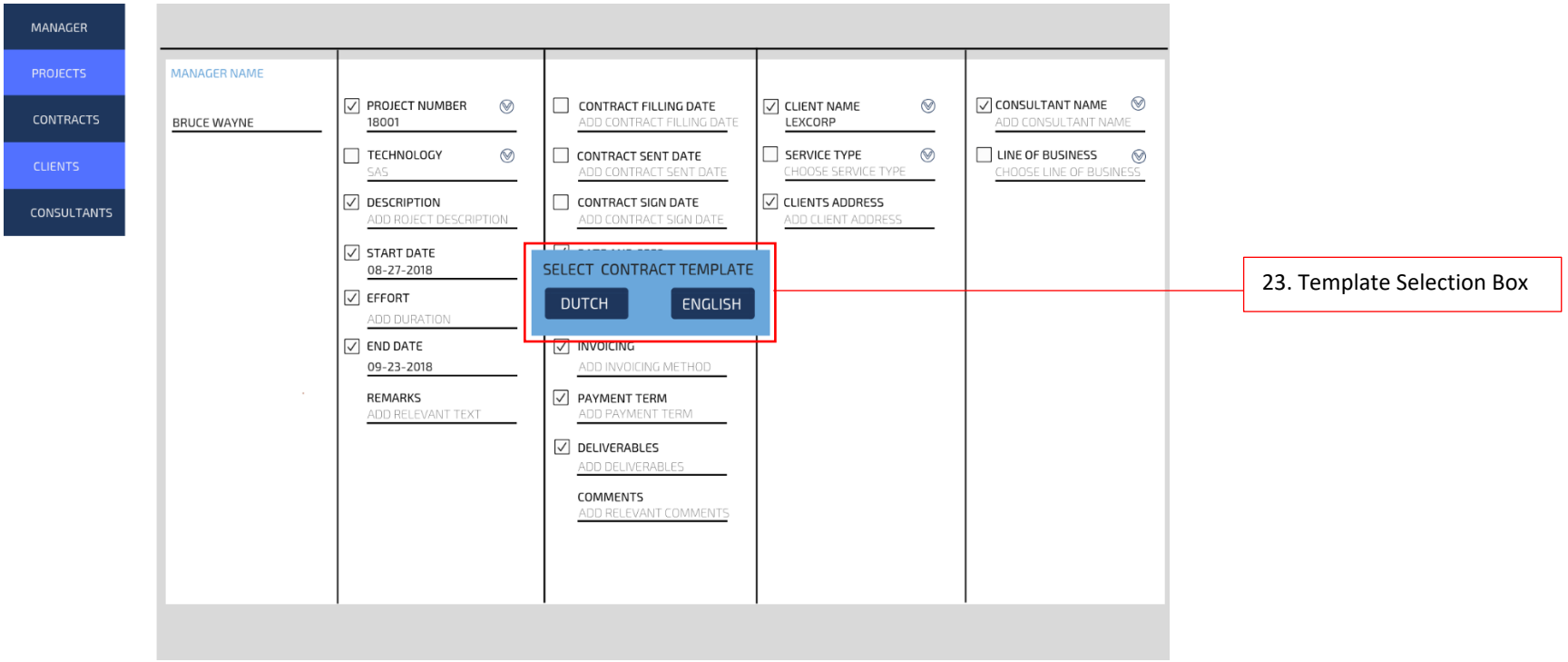


Fig.4

ID	Mock-up Breakdown Structure- Fig.4	Functional Description
23.	Template Selection Box	Template Selection Box is an interactive medium to store Form Fields into two arrangements namely, DUTCH and ENGLISH.

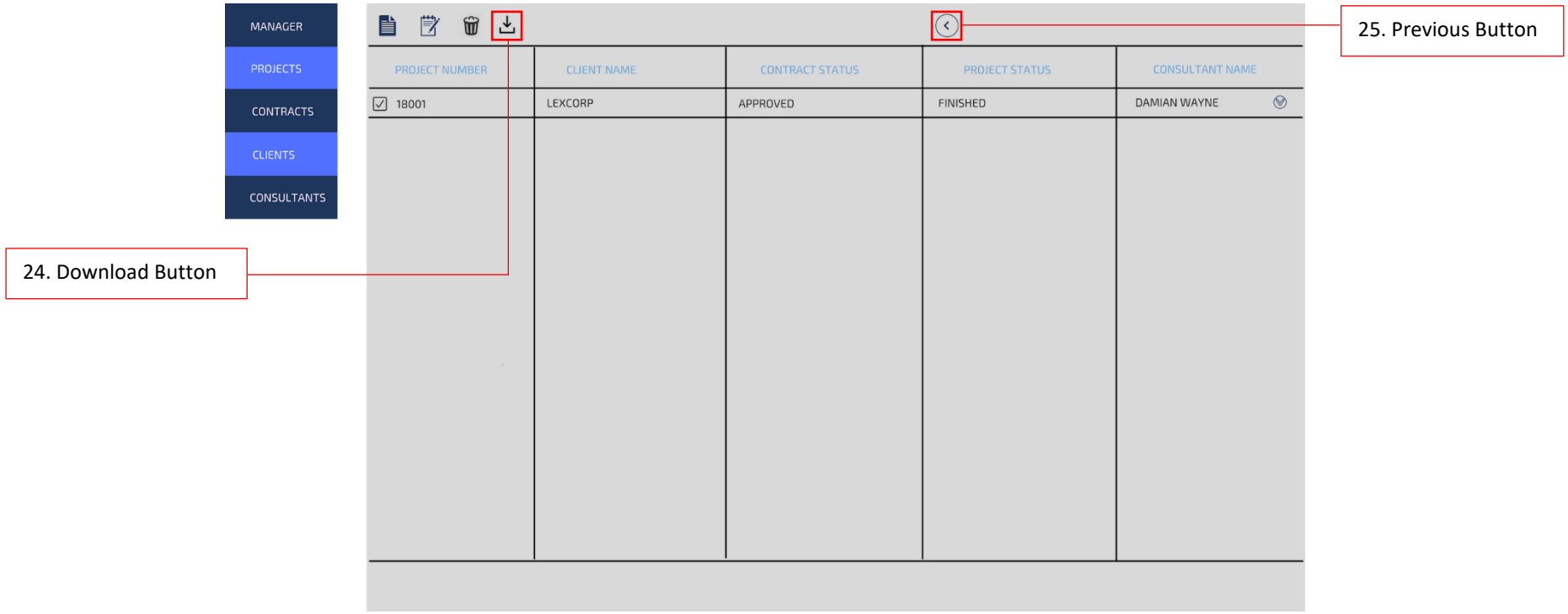


Fig.5

ID	Mock-up Breakdown Structure- Fig.5	Functional Description
24.	Download Button	Download Button is an interactive medium to output specific arrangement of Form Fields.
25.	Previous Button	Previous Button is an interactive medium to change Overviews.

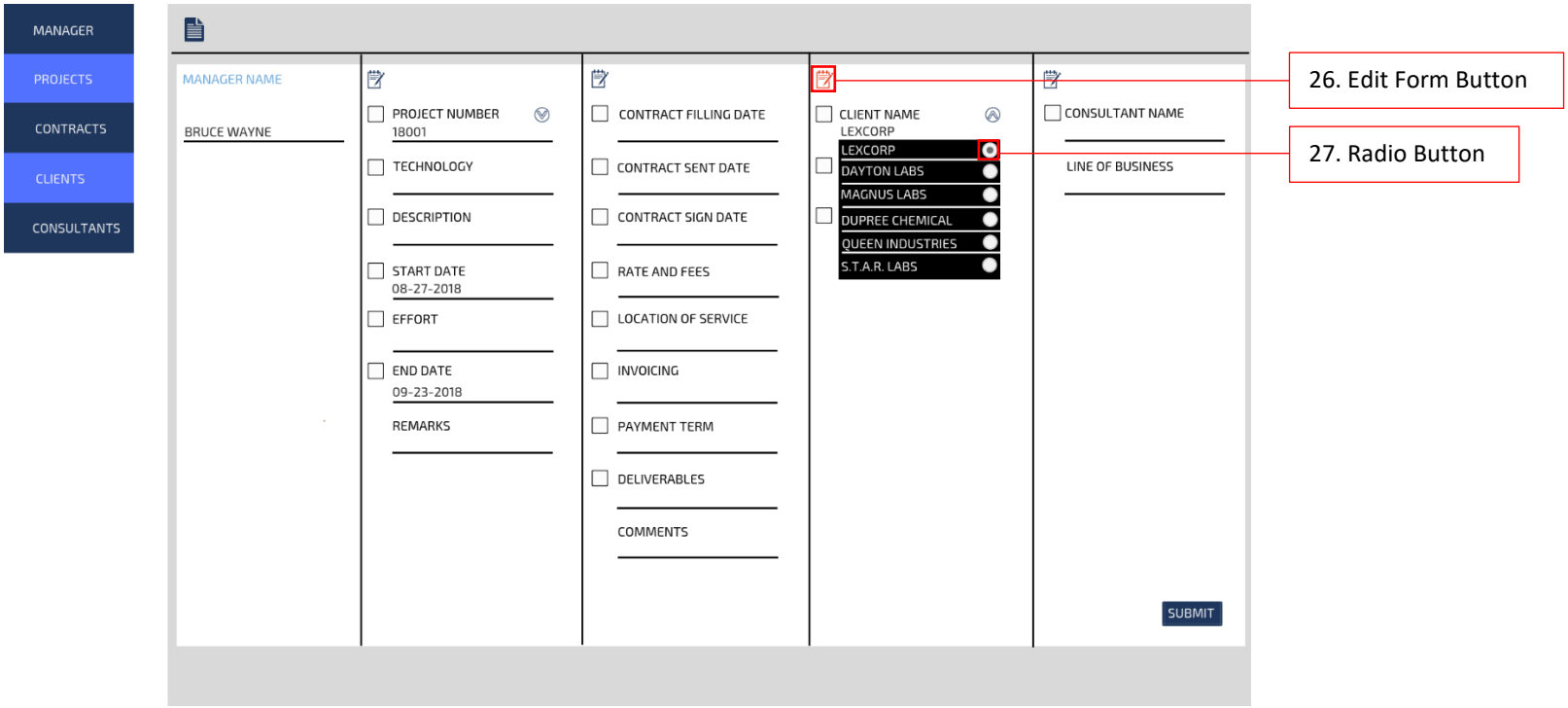


Fig.6

ID	Mock-up Breakdown Structure- Fig.6	Functional Description
26.	Edit Form Button	Edit Form Button is an interactive medium to change Forms.
27.	Radio Button	Radio Button is an interactive medium to select one input from a set of pre-defined inputs for certain Dropdown Menus.

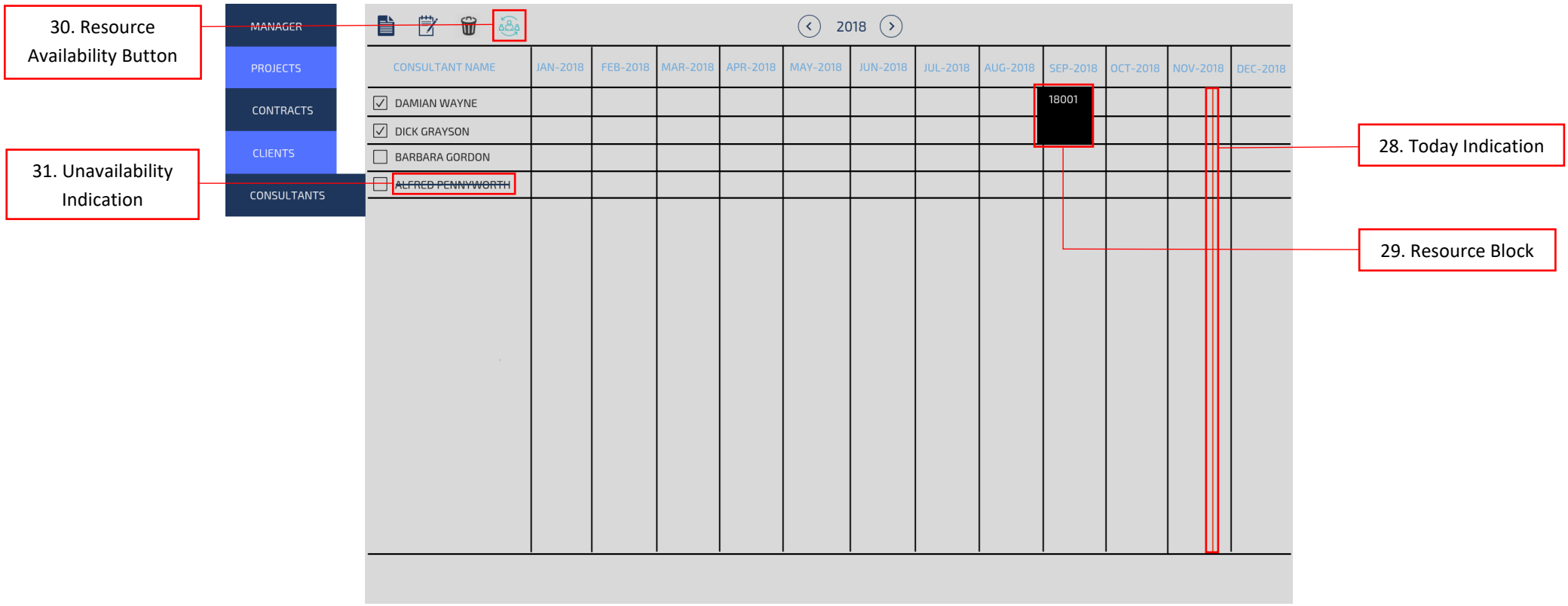


Fig.7

ID	Mock-up Breakdown Structure- Fig.7	Functional Description
28.	Today Indication	Today Indication is an indicative type medium to represent current date in the monthly calendar view.
29.	Resource Block	Resource Block is an indicative type medium to represent duration of Projects and the Consultants assigned.
30	Resource Availability Button	Resource Availability Button is an interactive medium to indicate the availability and unavailability of a Consultant.
31.	Unavailability Indication	Unavailability Indication is an indicative medium used to represent the current unavailability of a Consultant.

5. ERD Association of wireframe

Database Model Associations are the connections between the components in Web-Application Wireframe that correspond to the components of Form Fields and Overviews in Database Model. Table below defines these Database Model Associations.

ID.	Field in Web-Application Wireframe	Type	Table in Database Model	Variable Name - Type
1.	Project Number	Form Fields, Overviews	Projects	Project Number – Integer(10)
2.	Technology	Form Fields	Technology	Technology – Varchar(255)
3.	Description	Form Fields	Projects	Description – Longtext
4.	Start Date	Form Fields	Projects	Start Date – date
5.	Effort	Form Fields	Projects	Duration/Effort – Integer(10)
6.	End Date	Form Fields	Projects	End Date – date
7.	Remarks	Form Fields	Projects	Remarks – Longtext
8.	Contract Filling Date	Form Fields	Contracts	Contract Filling In Date – date
9.	Contract Sent Date	Form Fields	Contracts	Contract Sent Date – date
10.	Contract Sign Date	Form Fields	Contracts	Contract Sign Date – date
11.	Rate and Fees	Form Fields	Contracts	Rate and Fees – Longtext
12.	Location of Service	Form Fields	Contracts	Location of Service – Varchar(255)
13.	Invoicing	Form Fields	Contracts	Invoicing – Longtext
14.	Payment Term	Form Fields	Contracts	Payment Term – Longtext
15.	Delivery	Form Fields	Contracts	Delivery – Longtext
16.	Comments	Form Fields	Contracts	Comments – Varchar(255)
17.	Client Name	Form Fields, Overviews	Clients	Client Name – Varchar(255)
18.	Service Type	Form Fields, Overviews	Service Type	Service Type – Varchar(255)
19.	Clients Address	Form Fields, Overviews	Clients	Client Address – Varchar(255)
20.	Consultant’s Name	Form Fields, Overviews	Consultants	Consultant Name – Varchar(255)
21.	Line of Business	Form Fields, Overviews	Line of Business	Line of Business – Varchar(255)
22.	Project Status	Overviews	Project Status	Status – Varchar(255)
23.	Contract Status	Overviews	Contract Approval Status	Status – Varchar(255)
24.	Contract Number	Overviews	Extensions	ID – Integer(10)

6. Functional Requirements Mapping

Table below provides the functional requirements planned with Web-Application Wireframe along with their respective mock-up attachments.

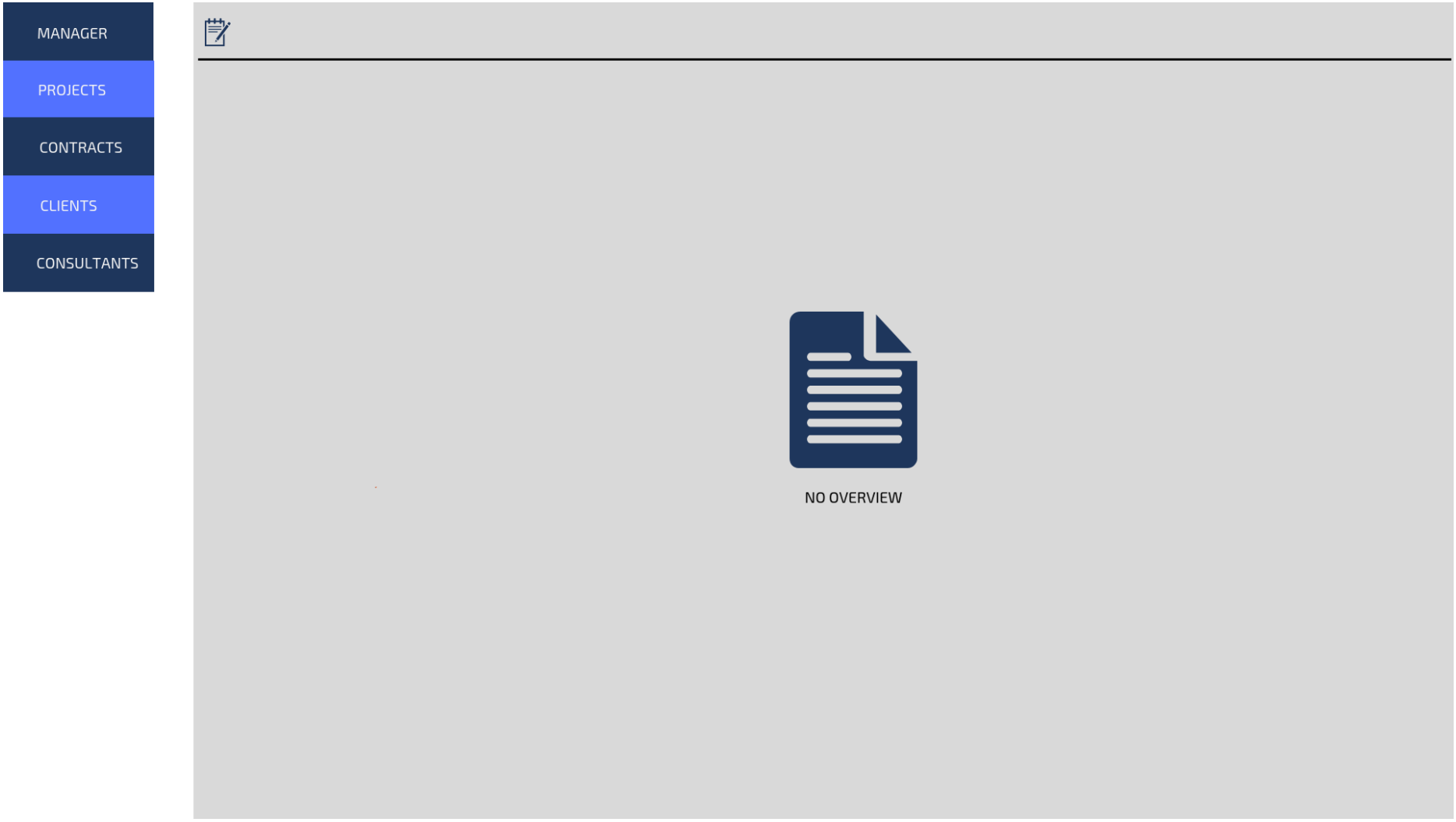
Functional Req. ID	**Process Description** • **Steps Involved**	**Mock-up**
1.	Starting point. • No information stored. • Click on Create Button.	SCREEN 1
2.	Assign the appropriate Project Number • Click on Dropdown Button in Form Field: Project Number. • Click on Add Button and Subtract Button until Allowed Indication on both Dropdown Menu inputs.	SCREEN 2, SCREEN 3, SCREEN 4, SCREEN 5
3.	Change Forms. • Click on Form Button: NEXT and BACK.	SCREEN 5, SCREEN 6, SCREEN 7, SCREEN 8
4.	Store and view entries with Forms into database. • Type in Form Fields. • Click Form Button: SUBMIT.	SCREEN 8, SCREEN 9, SCREEN 10, SCREEN 13
5.	Enter one Client. • Click Dropdown Button in Form Field: Client Name. • Click on Radio Button of input. • Click Form Button: SUBMIT.	SCREEN 10, SCREEN 12
6.	Enter one or more Consultants. • Click Dropdown Button in Form Field: Consultant Name. • Fill Check Box of inputs. • Click Form Button: SUBMIT.	SCREEN 13
7.	View, remove and edit Clients Overviews. • Click Clients tab in Side Navigation. • Fill in Check Box of Clients Overviews. • Click Remove Button and Edit Button.	SCREEN 10, SCREEN 11
8.	View, remove, edit and indicate the availability, unavailability of Consultant(s). • Click Consultants tab in Side Navigation. • Fill in Check Box of Consultants Overviews. • Click Remove Button, Edit Button and Resource Availability Button.	SCREEN 13, SCREEN 14

9.	Change the year from default year, i.e. change present year to another year of Consultants Overviews. • Click Consultants tab in Side Navigation. • Click on Next Button and Previous Button.	SCREEN 14
10.	View, remove and edit Overviews. • Click Overview Button. • Fill in Check Box of Overviews. • Click Remove Button and Edit Button.	SCREEN 15, SCREEN 16
11.	Generate a Contract. • Click Create Button. • Fill in Check Box of Form Fields. • Click Generate Button. • Click DUTCH or ENGLISH in Template Selection Box. • Click on Overview Button. • Fill in Check Box of input. • Click Download Button.	SCREEN 17, SCREEN 18, SCREEN 19
12.	Upload an approved Contract. • Generate a Contract. • Click on Approve Button. • Click on Upload Button.	SCREEN 19, SCREEN 20.
13.	Start and finish a Project. • Upload an approved Contract. • Click on Next Button once to start. • Click on Next Button two more times to finish.	SCREEN 21, SCREEN 22, SCREEN 23, SCREEN 27
14.	Extend a Contract. • Start a Project. • Click Approve Button. • Click Upload Button. • Click Previous Button.	SCREEN 23, SCREEN 24, SCREEN 25
15.	View, delete and download extended Contracts Overviews. • Click Contracts tab. • Fill in Check Box of Overviews. • Click Remove Button and Download Button.	SCREEN 26

Table below provides the mock-ups of Web-Application Wireframe to illustrate the above functional requirements.

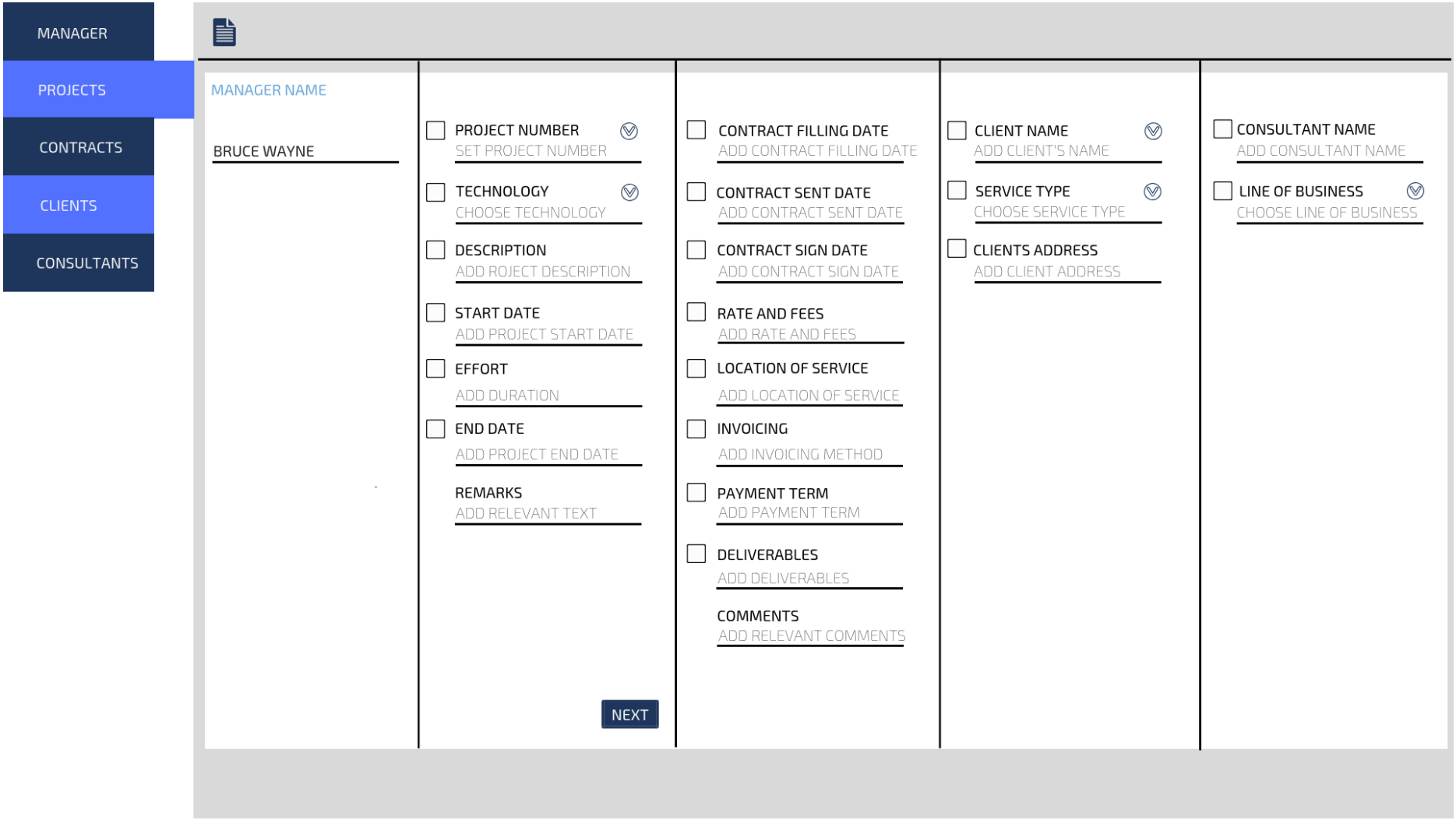
SCREEN 1

Functional Req. ID:	Functional Req. ID 1.



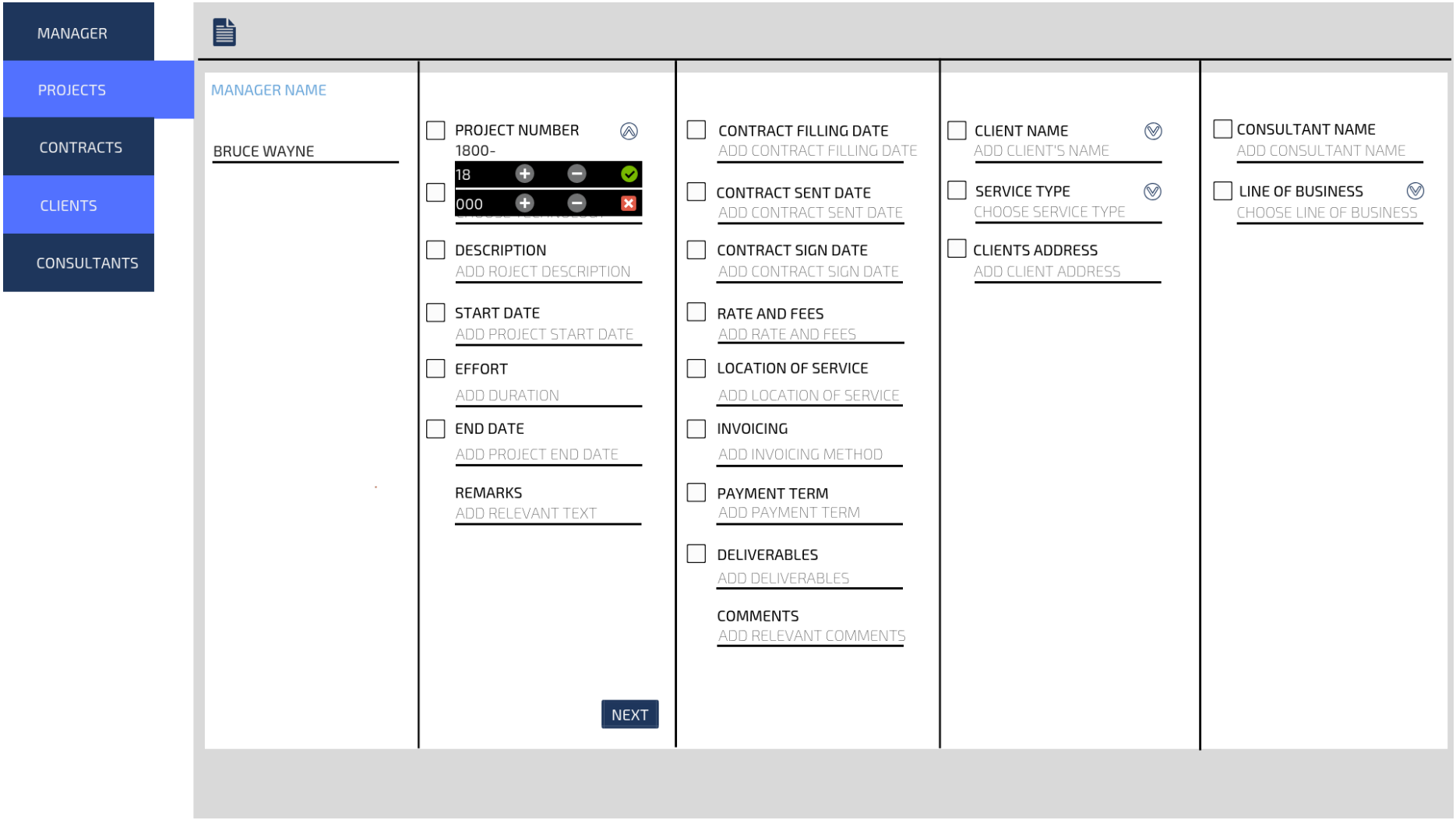
SCREEN 2

Functional Req. ID:	Functional Req. ID_2.



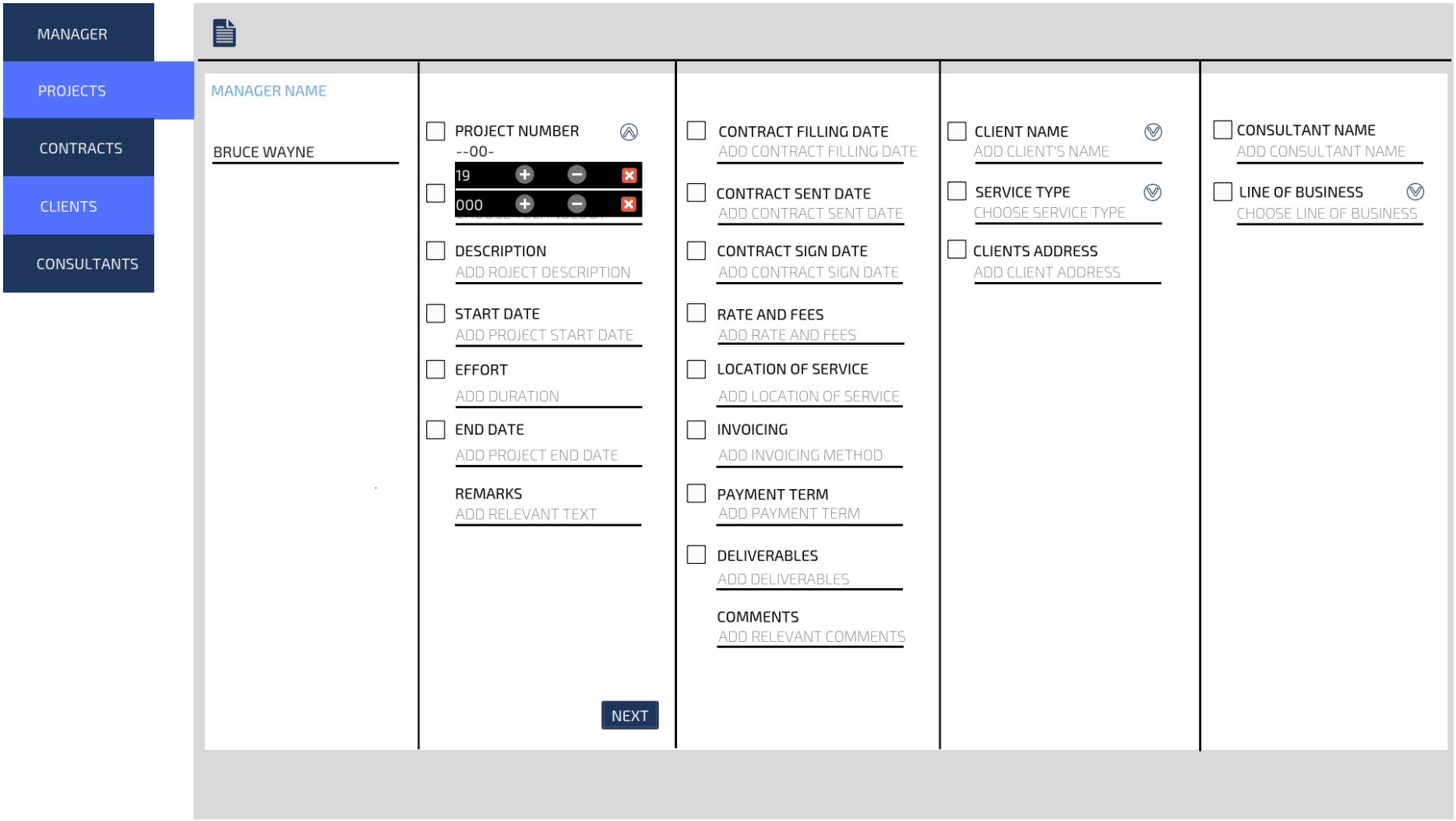
SCREEN 3

Functional Req. ID:	Functional Req. ID 2.



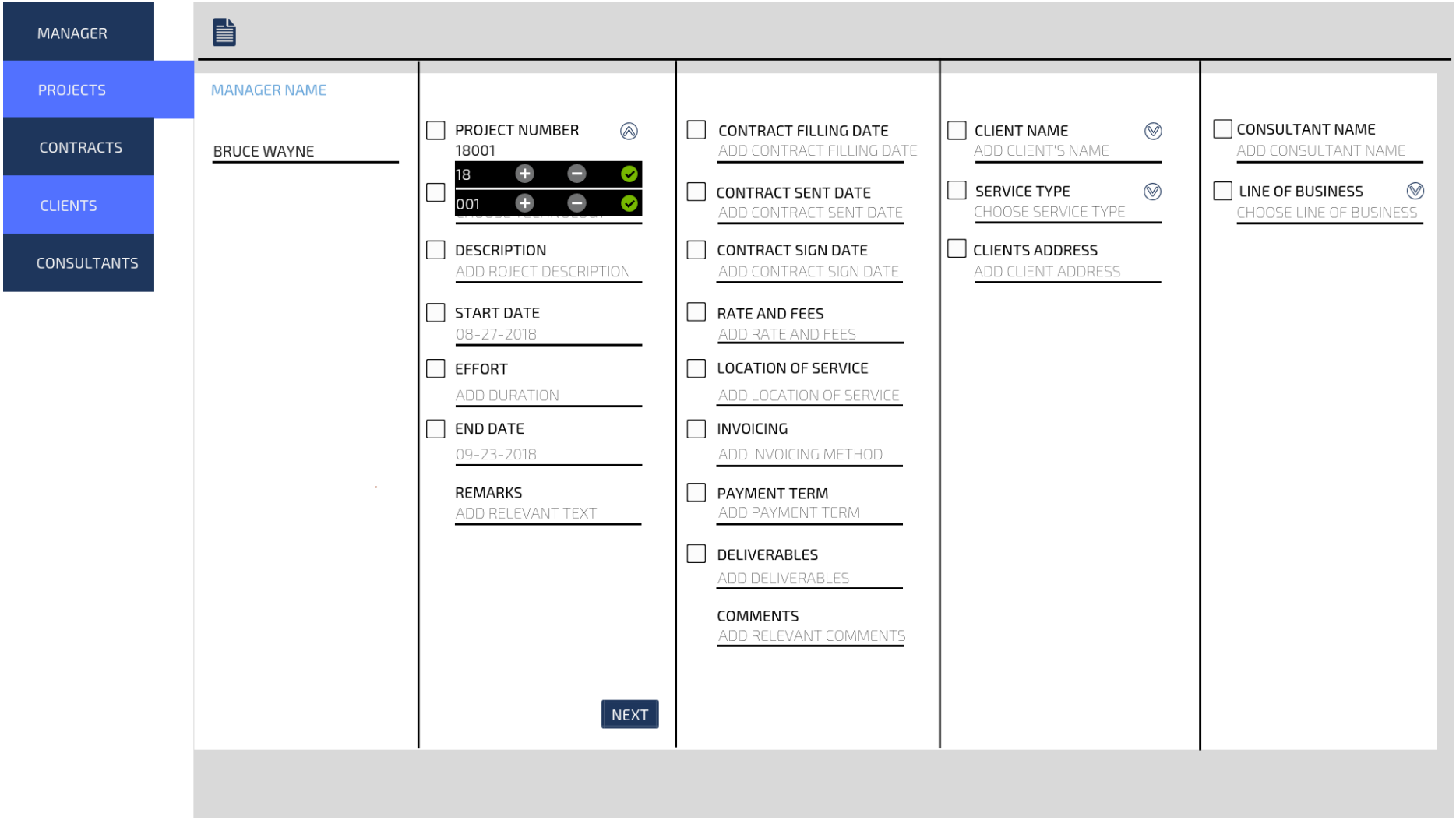
SCREEN 4

Functional Req. ID:	Functional Req. ID 2.



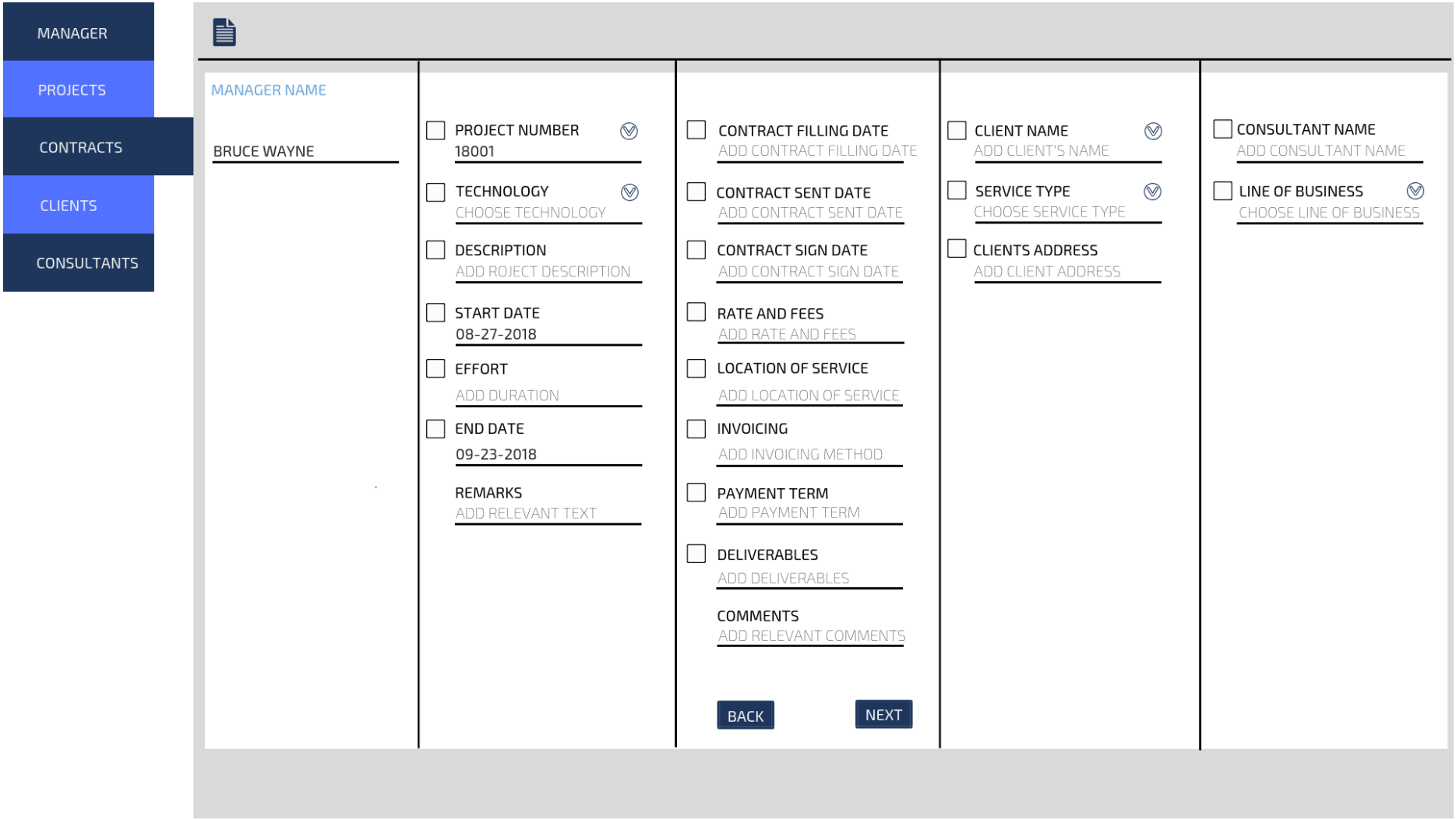
SCREEN 5

Functional Req. ID:	Functional Req. ID 2., Functional Req. ID 3.



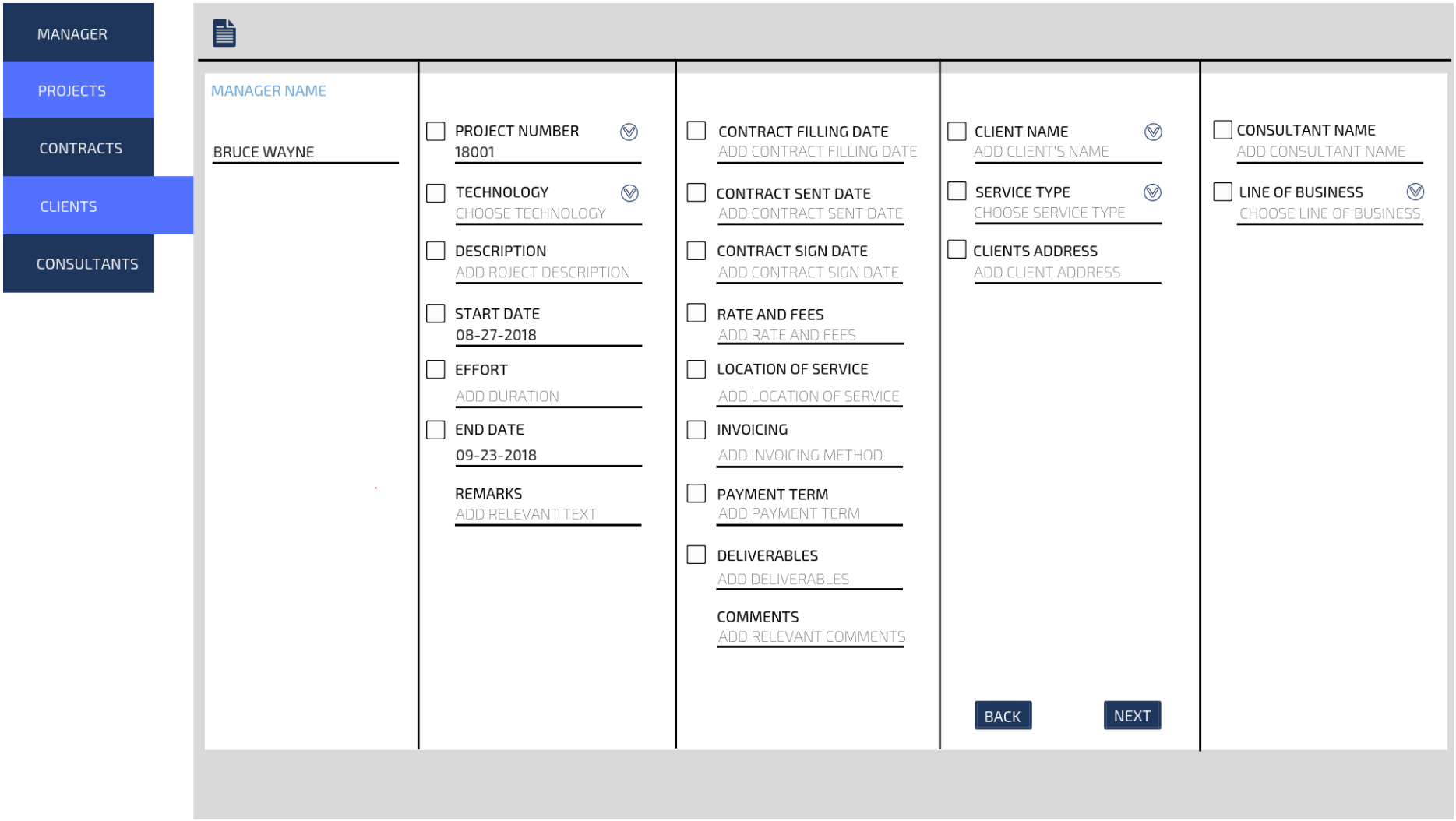
SCREEN 6

Functional Req. ID:	Functional Req. ID 3.



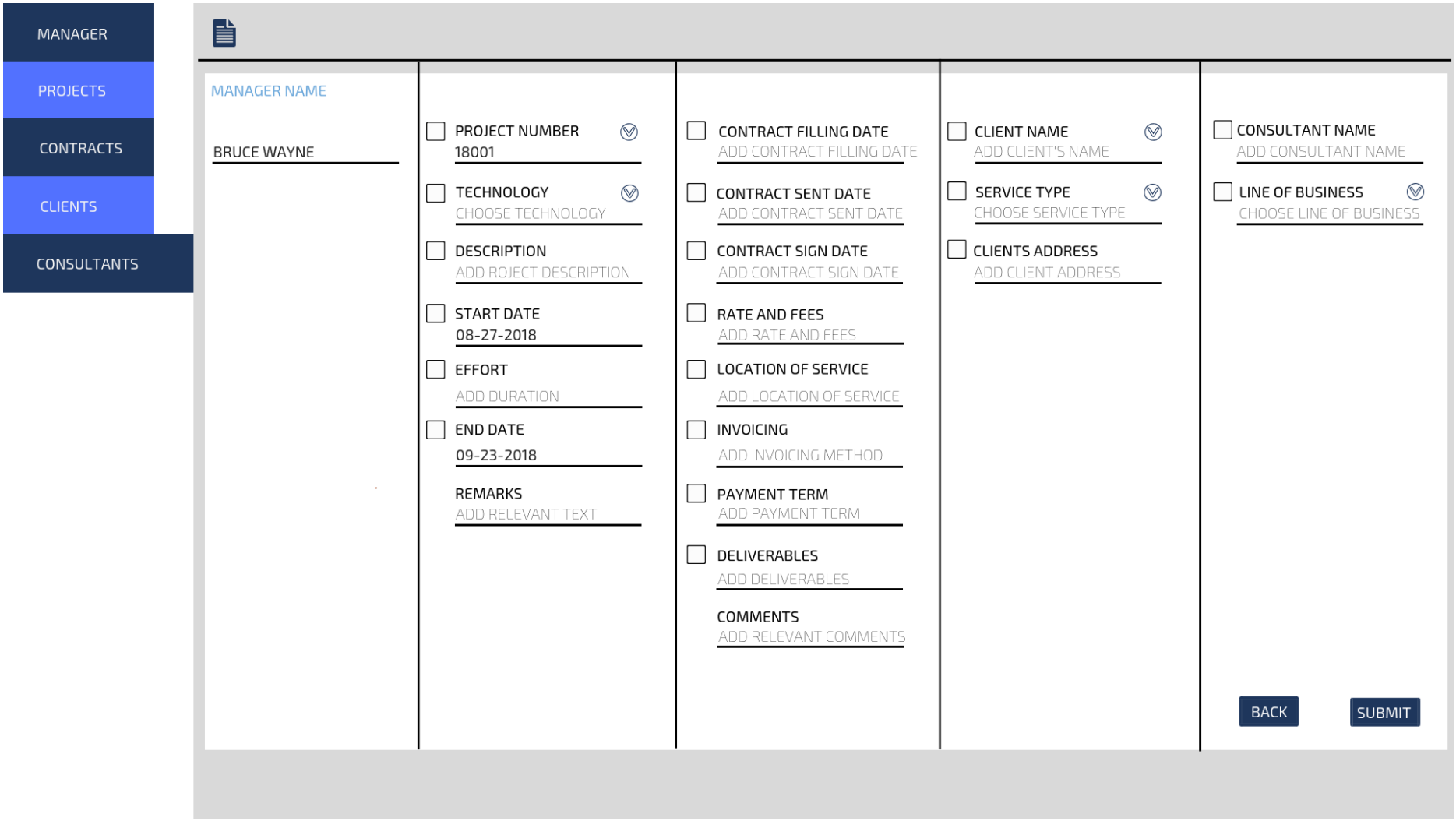
SCREEN 7

Functional Req. ID:	Functional Req. ID 3.



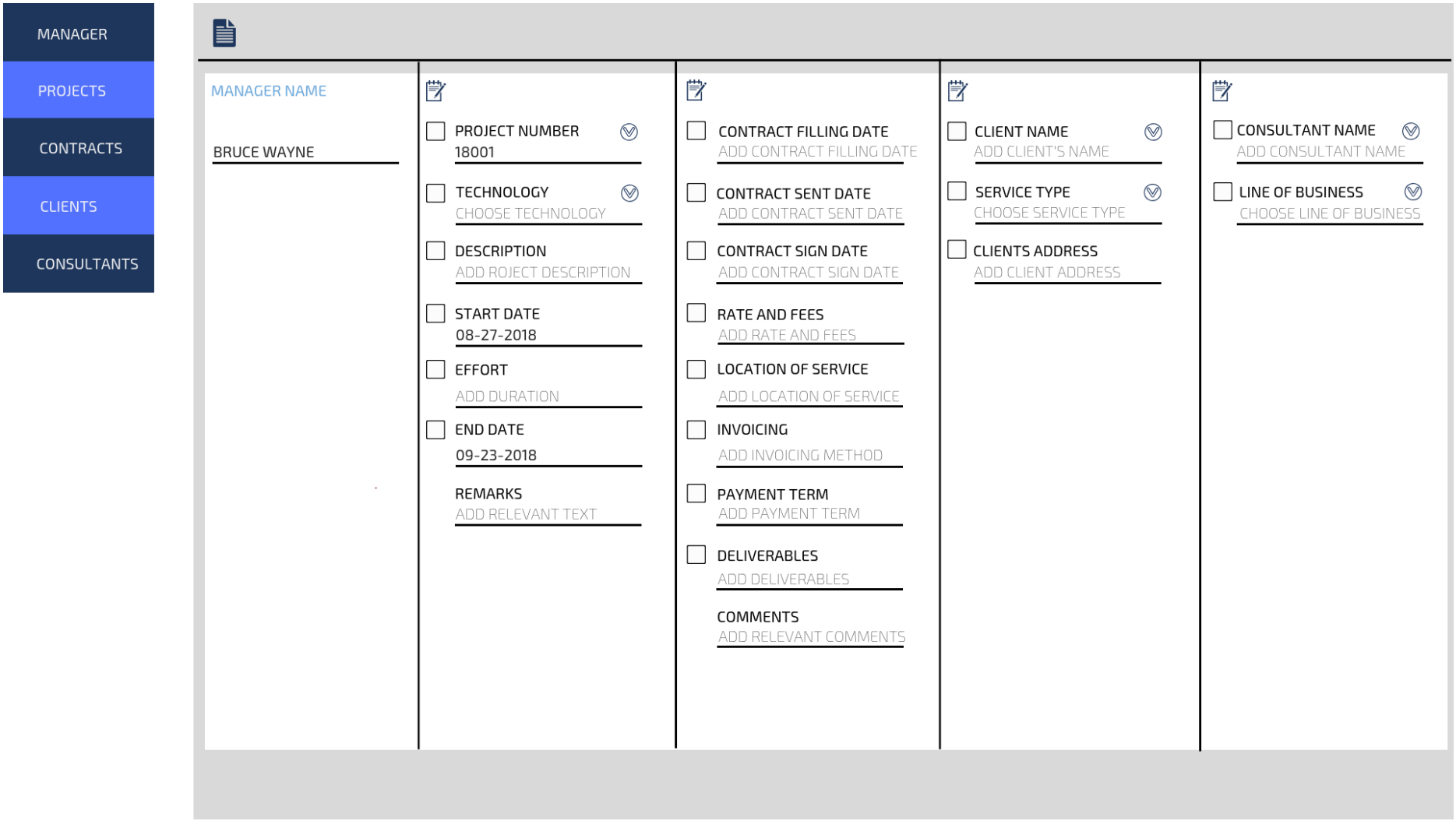
SCREEN 8

Functional Req. ID:	Functional Req. ID 3., Functional Req. ID 4.



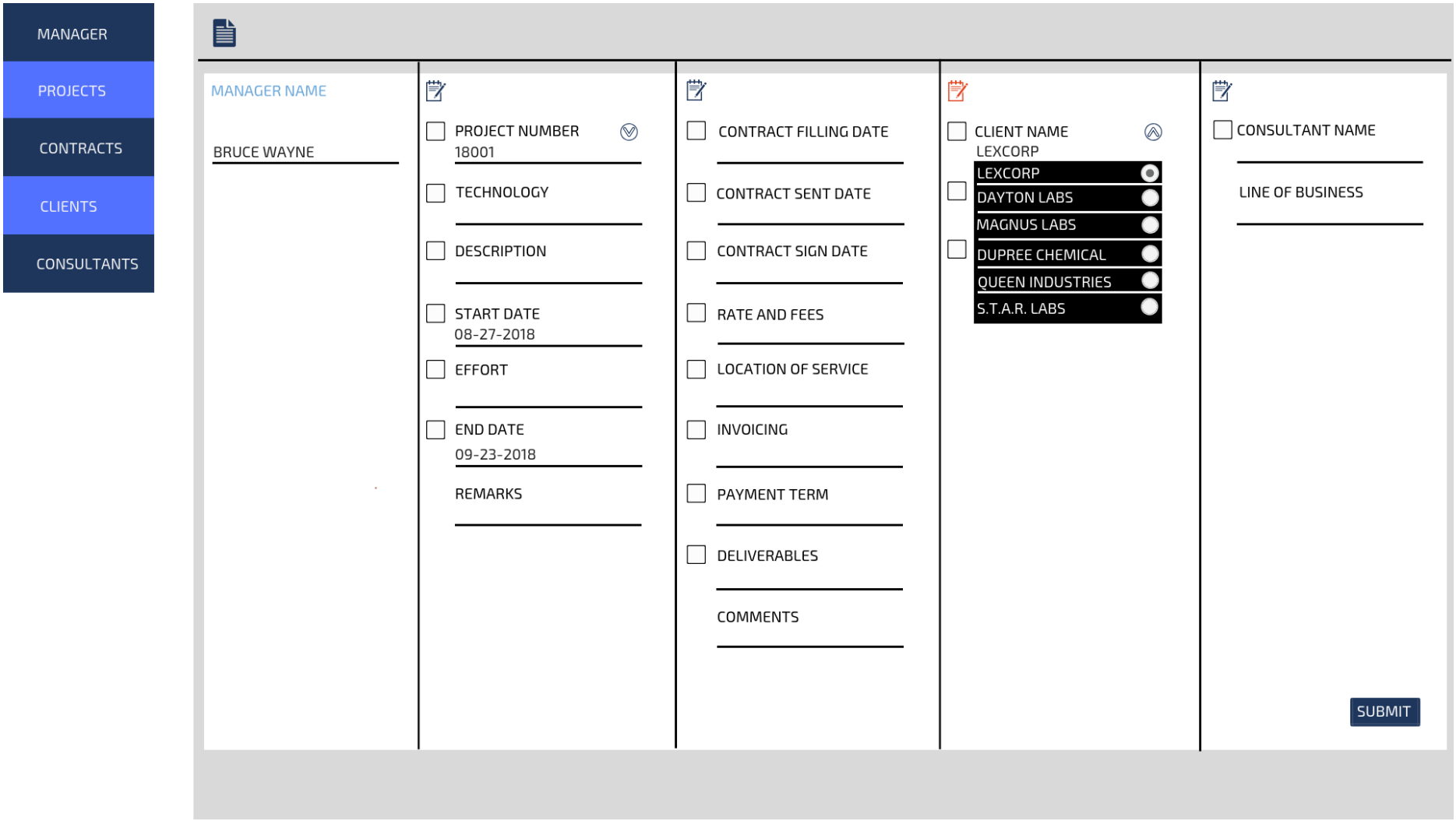
SCREEN 9

Functional Req. ID:	Functional Req. ID 4.



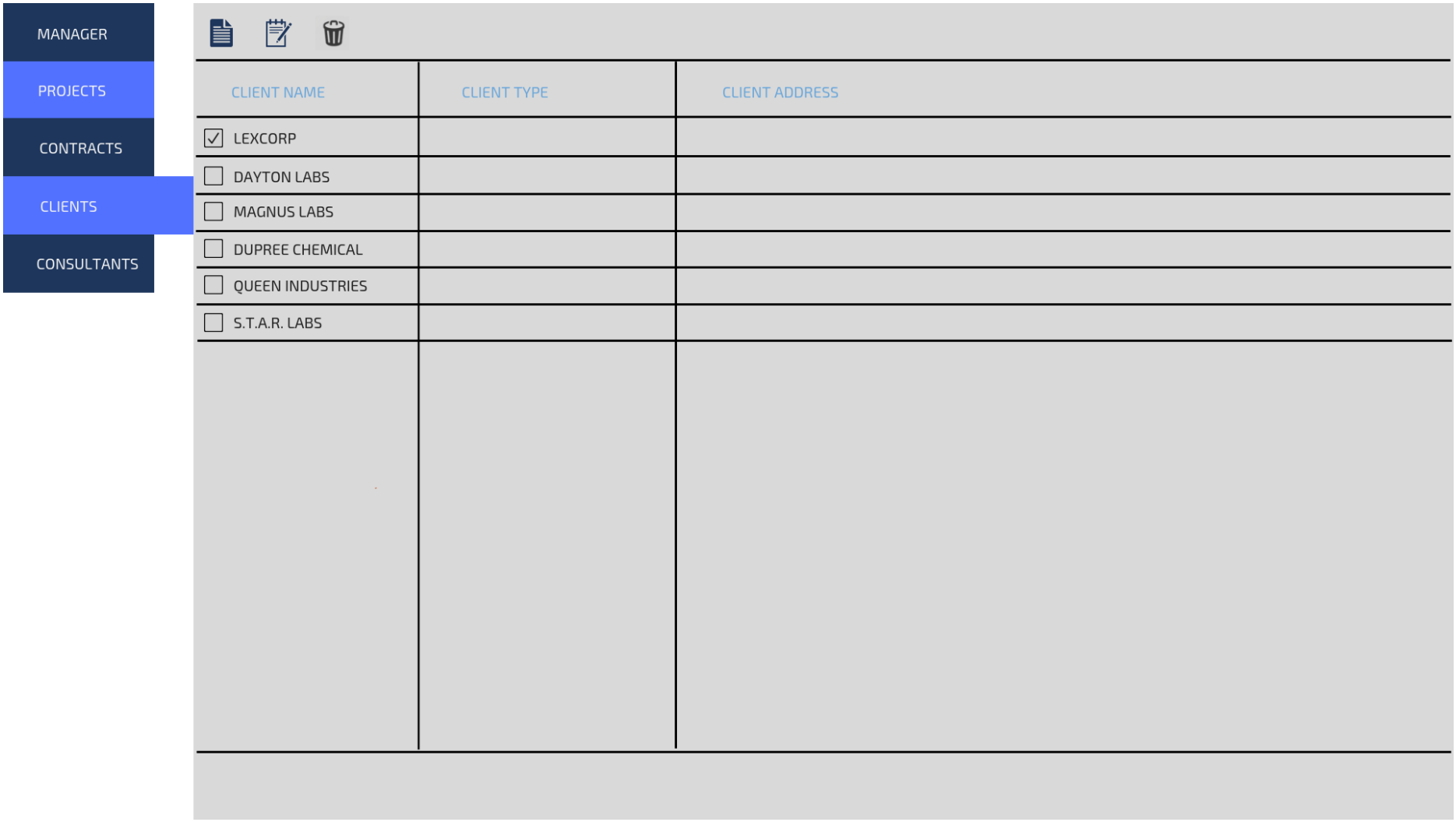
SCREEN 10

Functional Req. ID:	Functional Req. ID 4., Functional Req. ID 5., Functional Req. ID 7



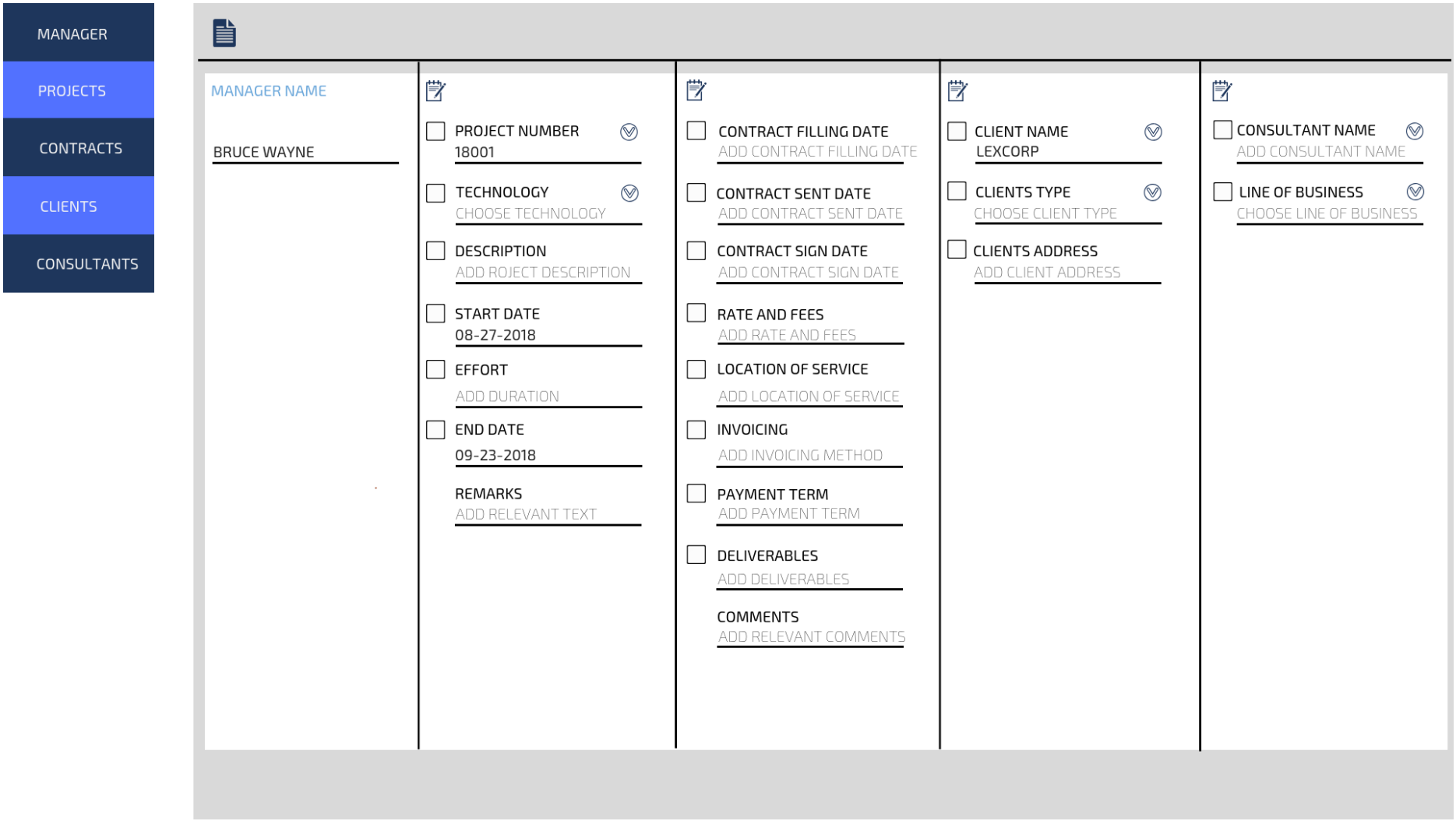
SCREEN 11

Functional Req. ID:	Functional Req. ID 7.



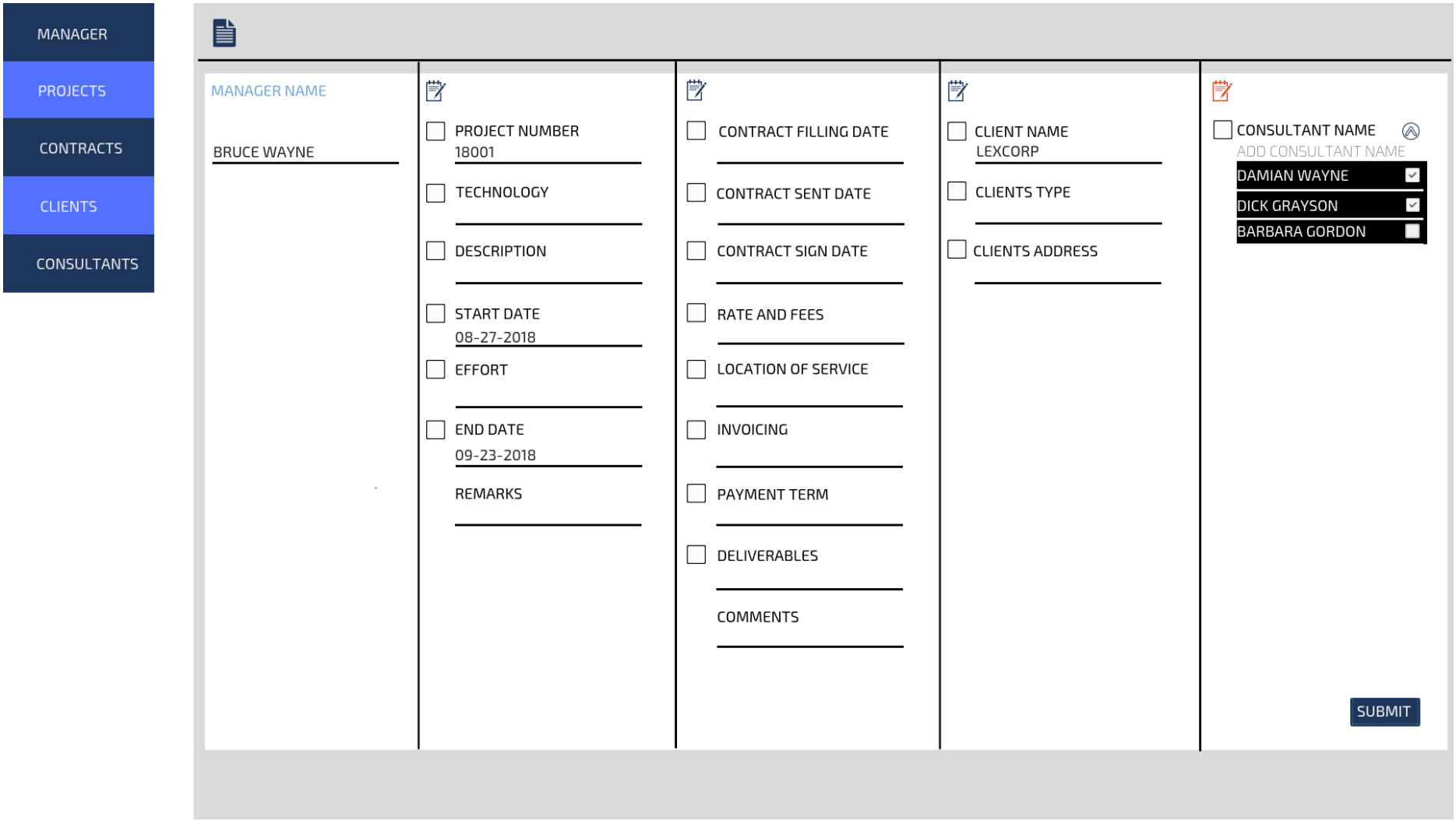
SCREEN 12

Functional Req. ID:	Functional Req. ID 5.



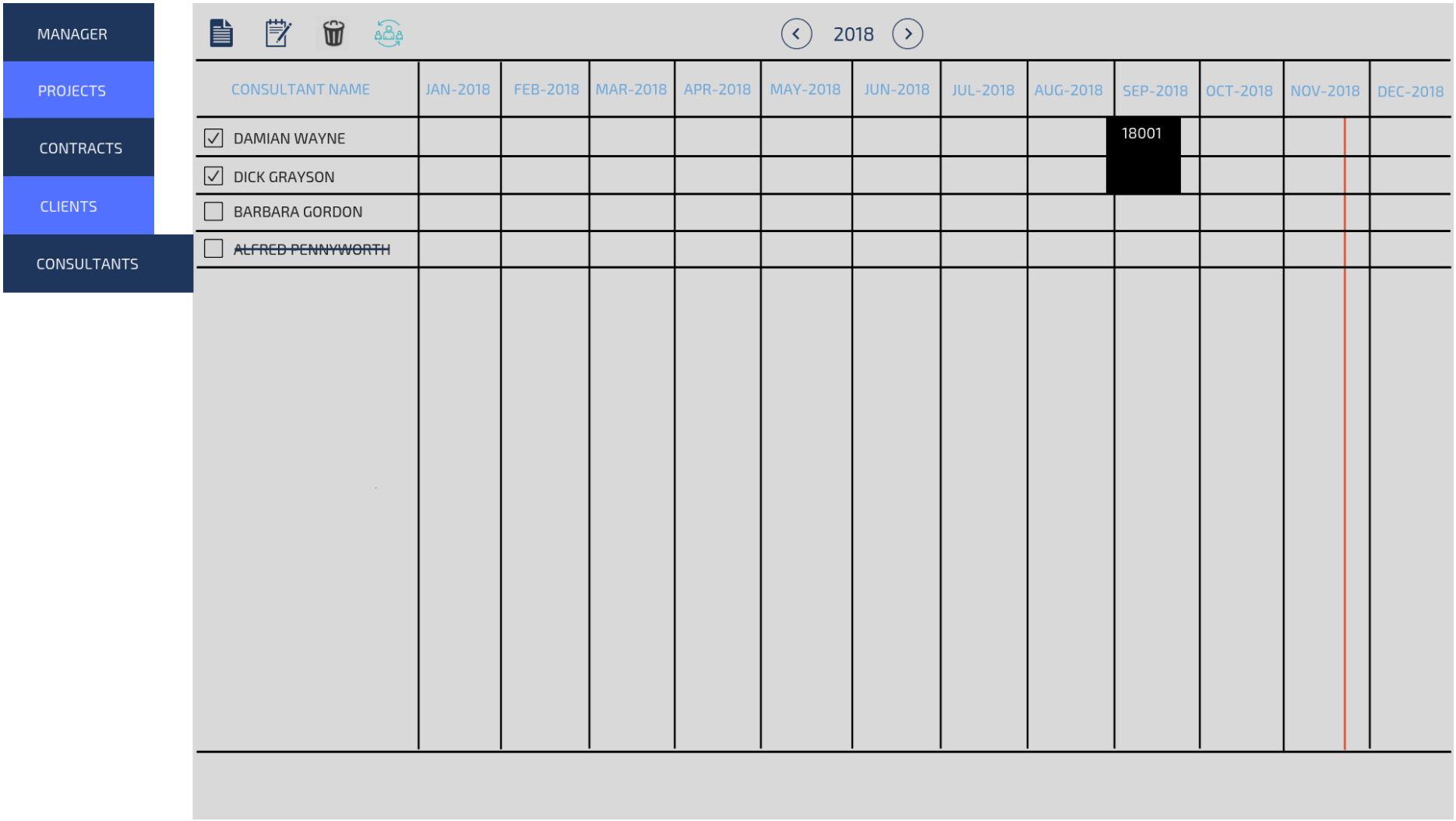
SCREEN 13

Functional Req. ID:	Functional Req. ID 4., Functional Req. ID 6., Functional Req. 8



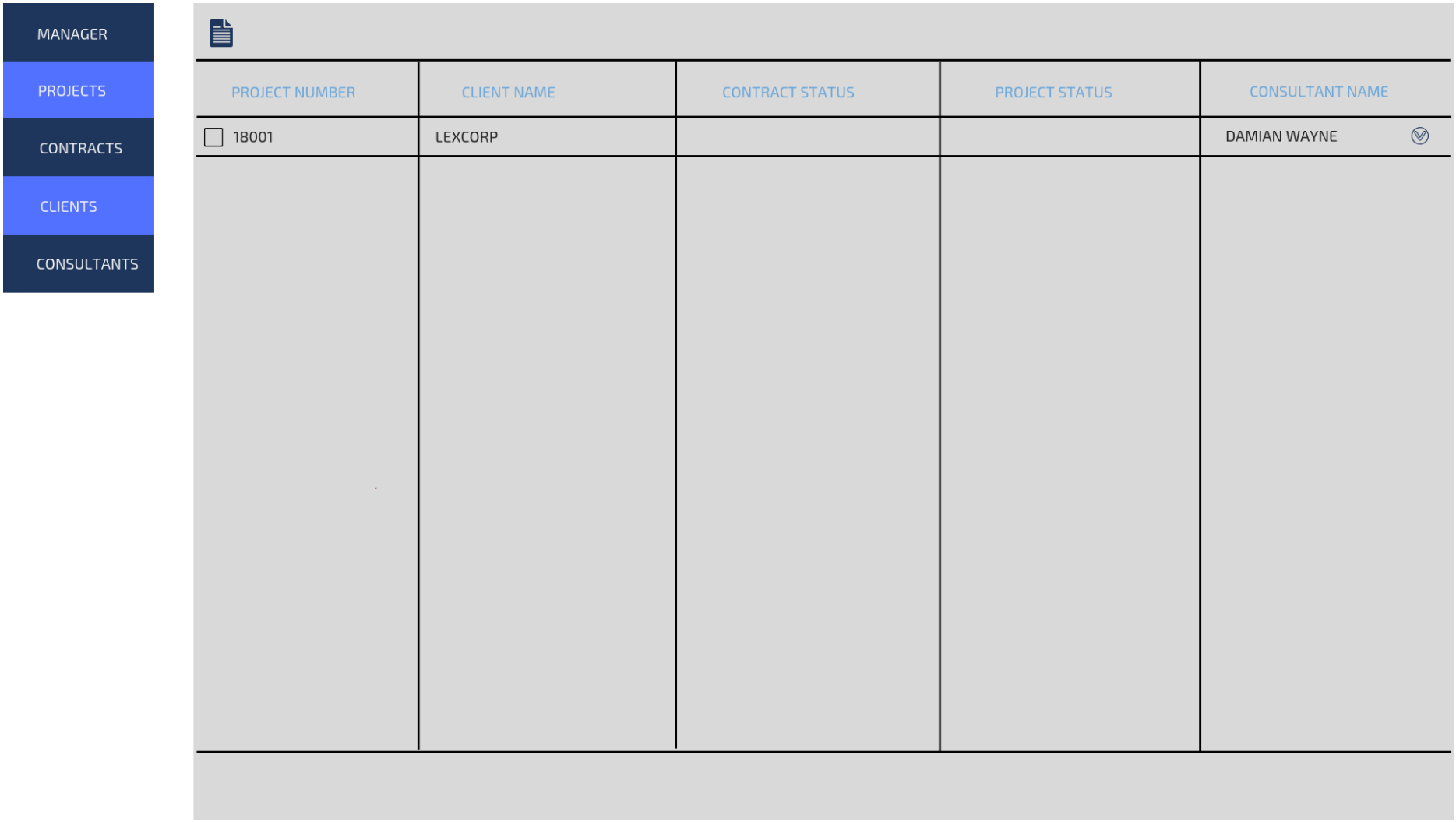
SCREEN 14

Functional Req. ID:	Functional Req. ID 8., Functional Req. ID 9.



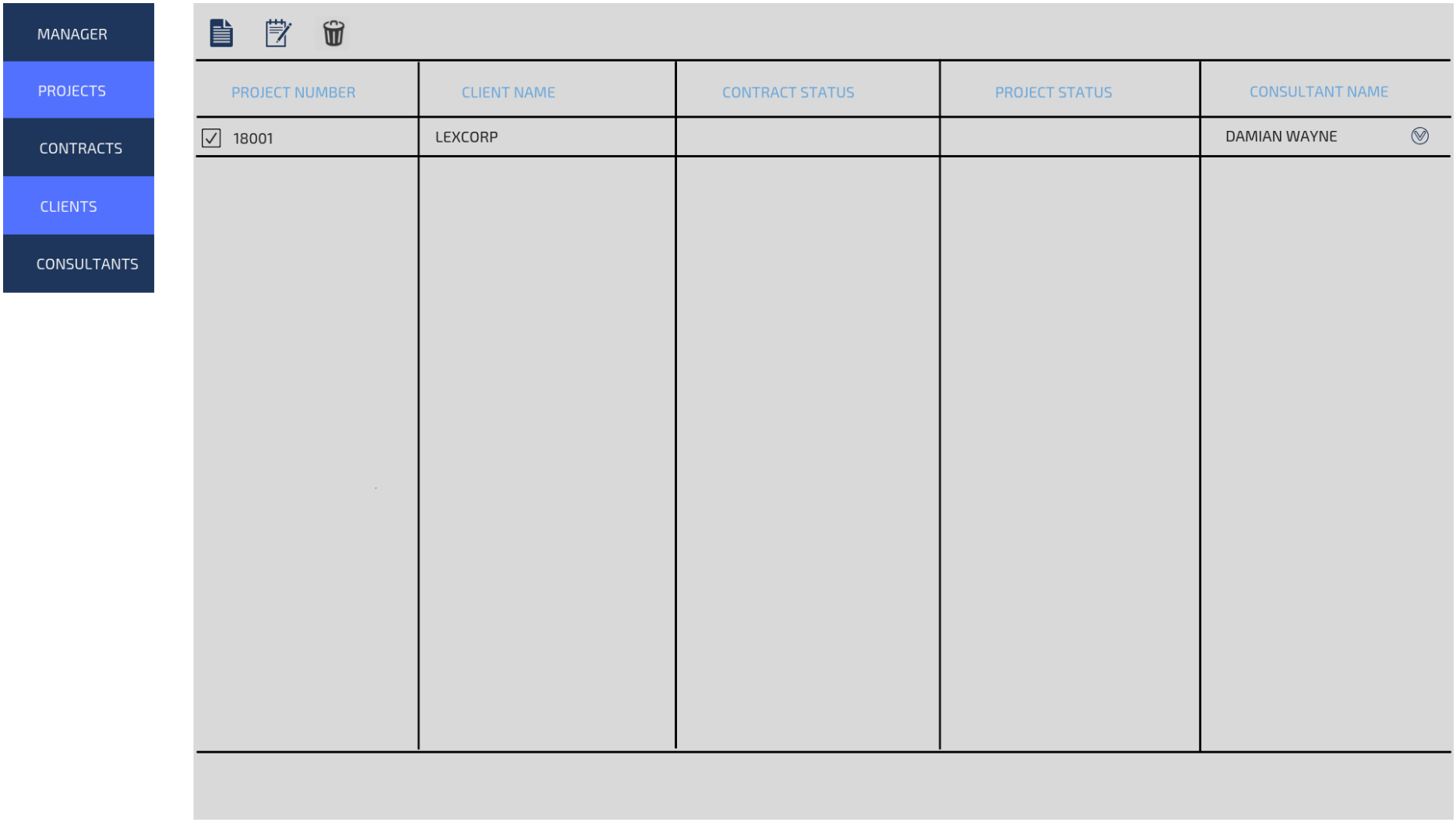
SCREEN 15

Functional Req. ID:	Functional Req. ID 10



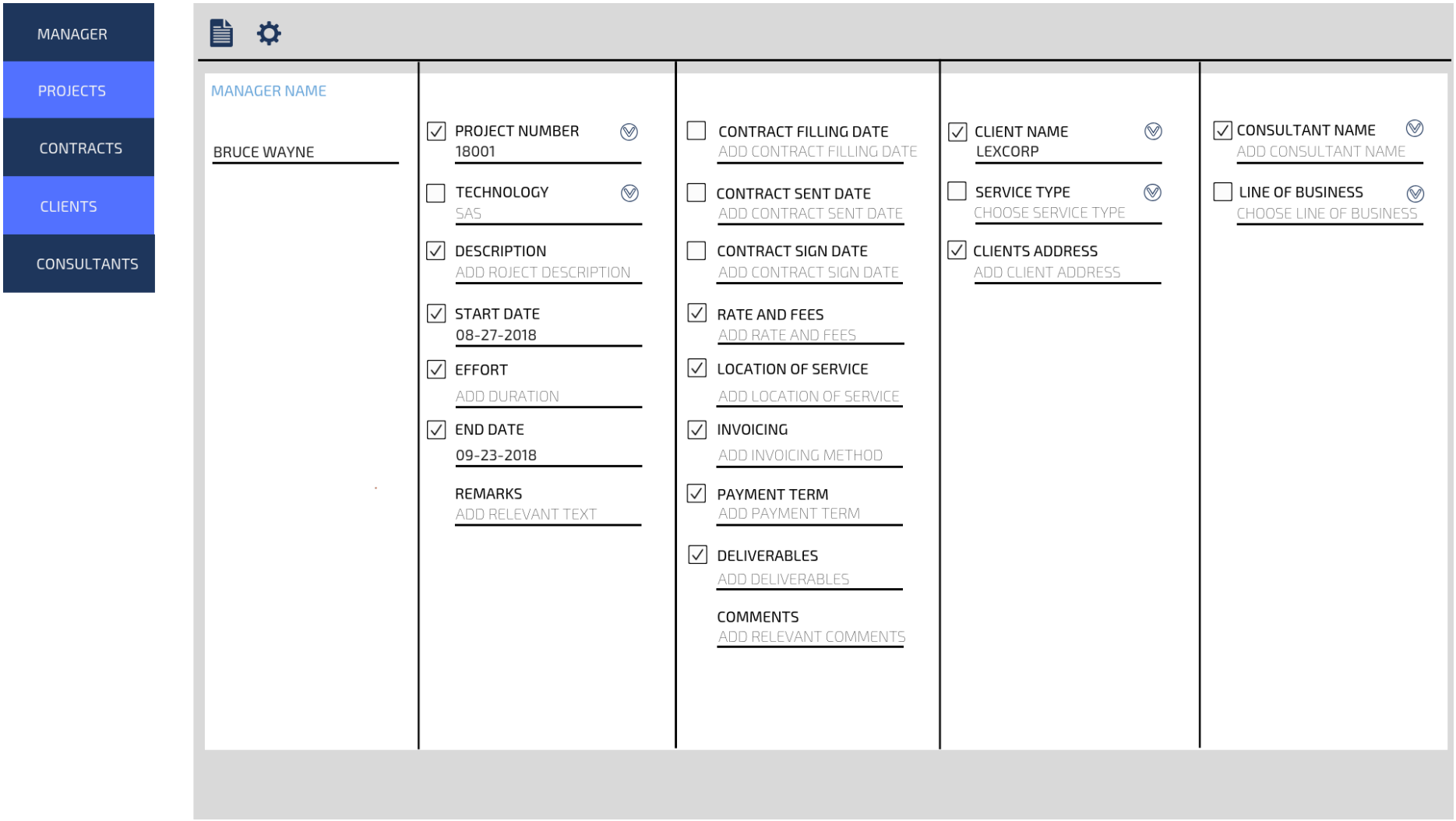
SCREEN 16

Functional Req. ID:	Functional Req. ID 10



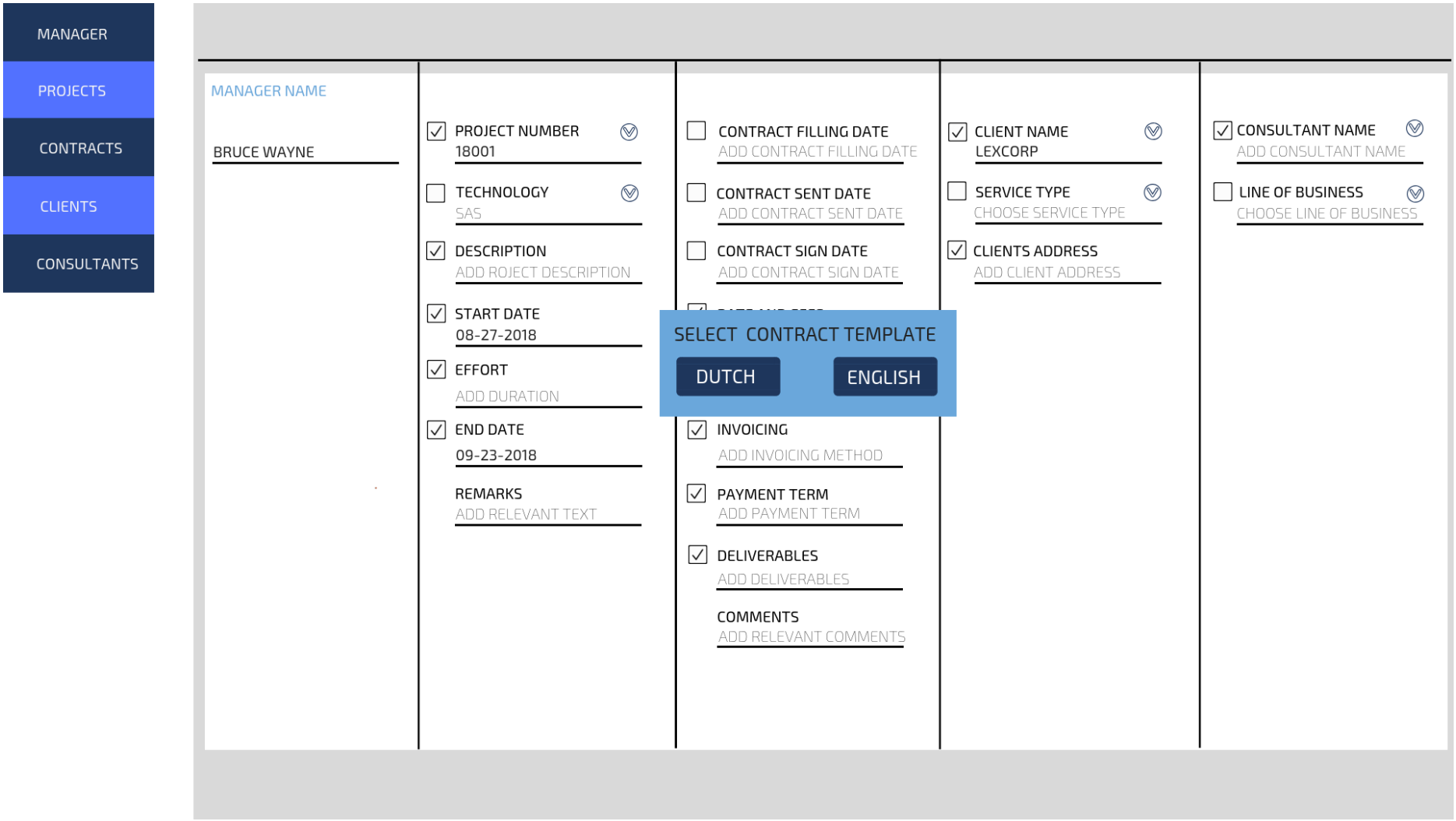
SCREEN 17

Functional Req. ID:	Functional Req. ID 11



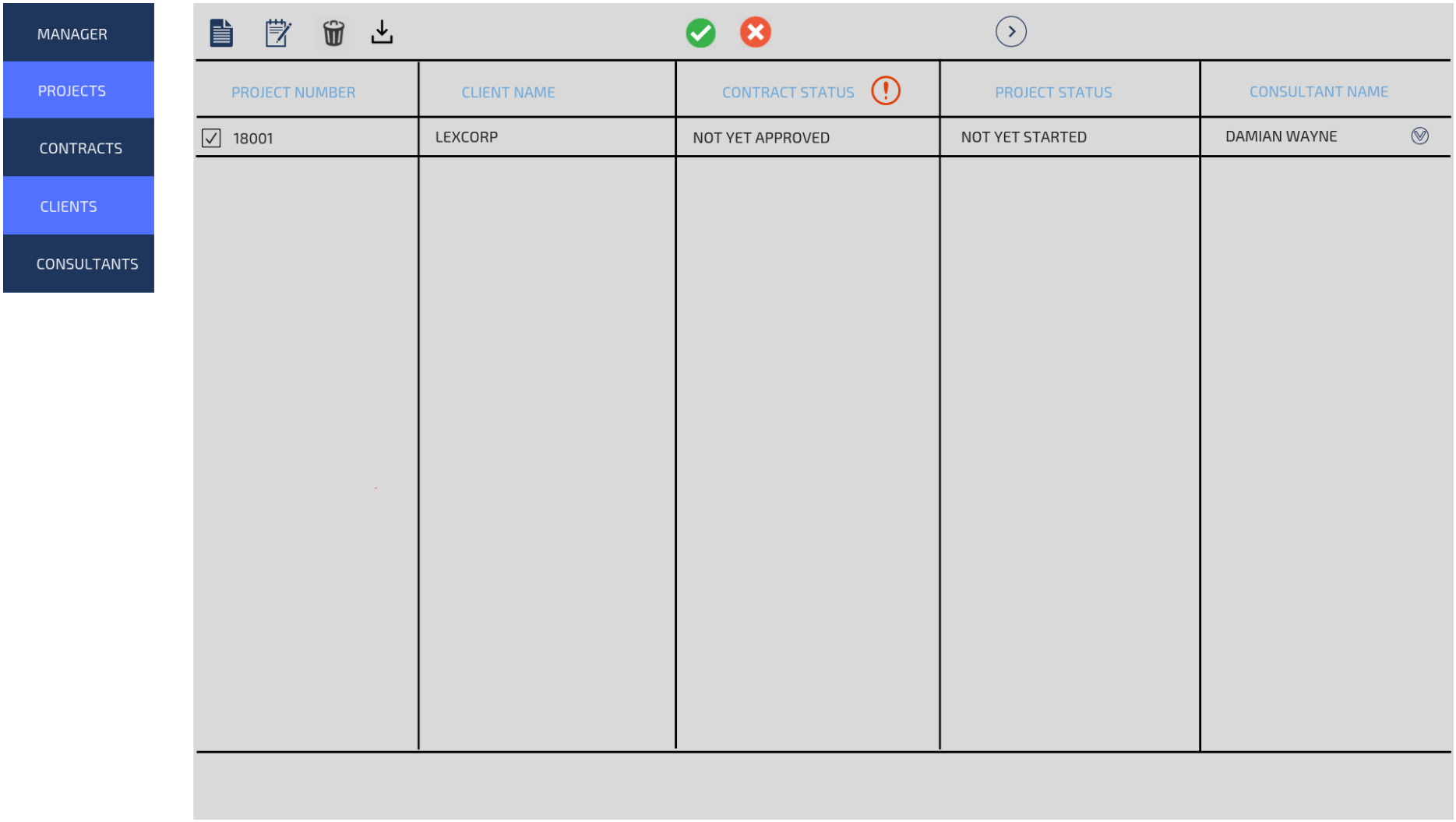
SCREEN 18

Functional Req. ID:	Functional Req. ID 11



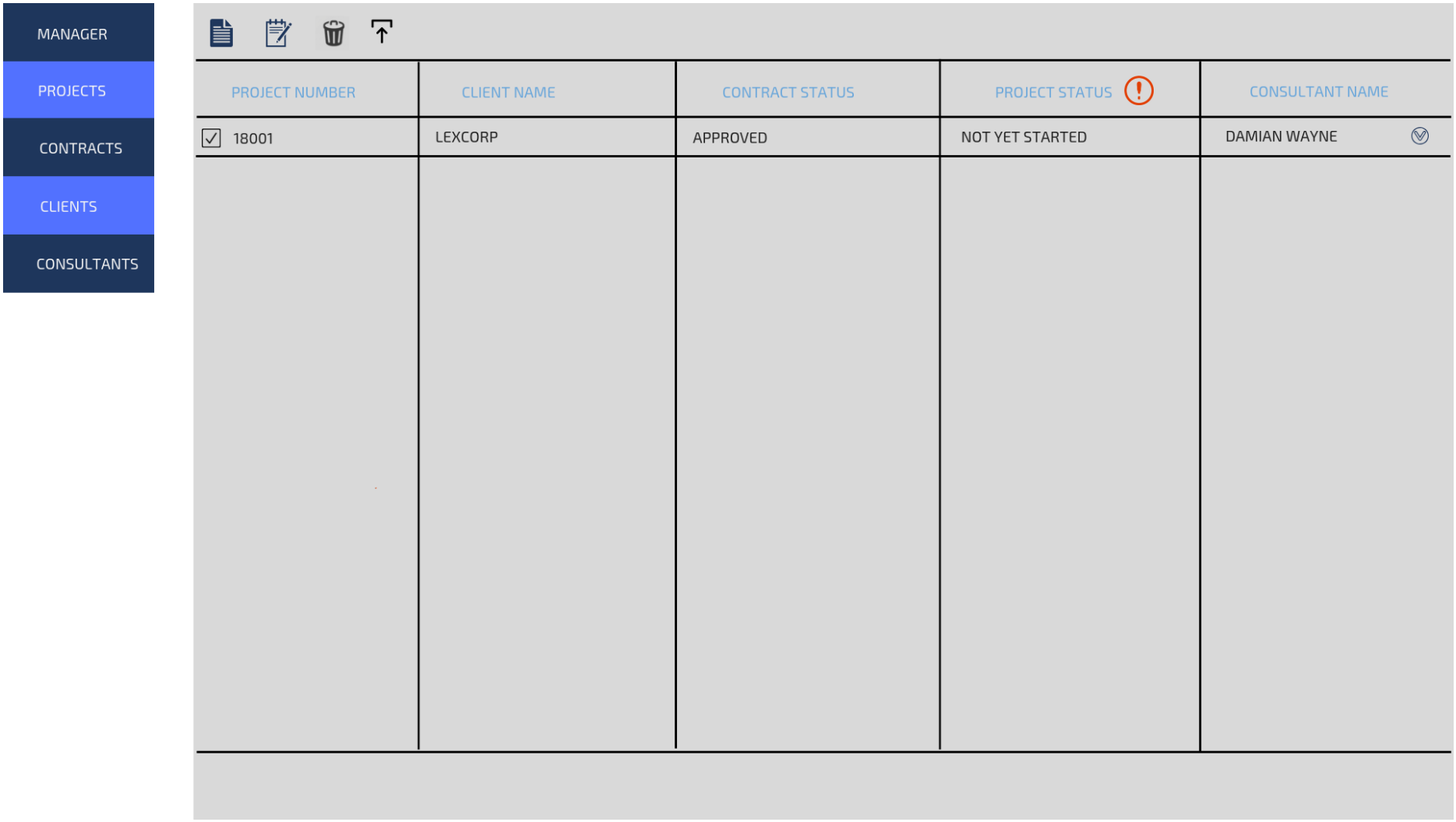
SCREEN 19

Functional Req. ID:	Functional req. ID 11., Functional req. ID 12.



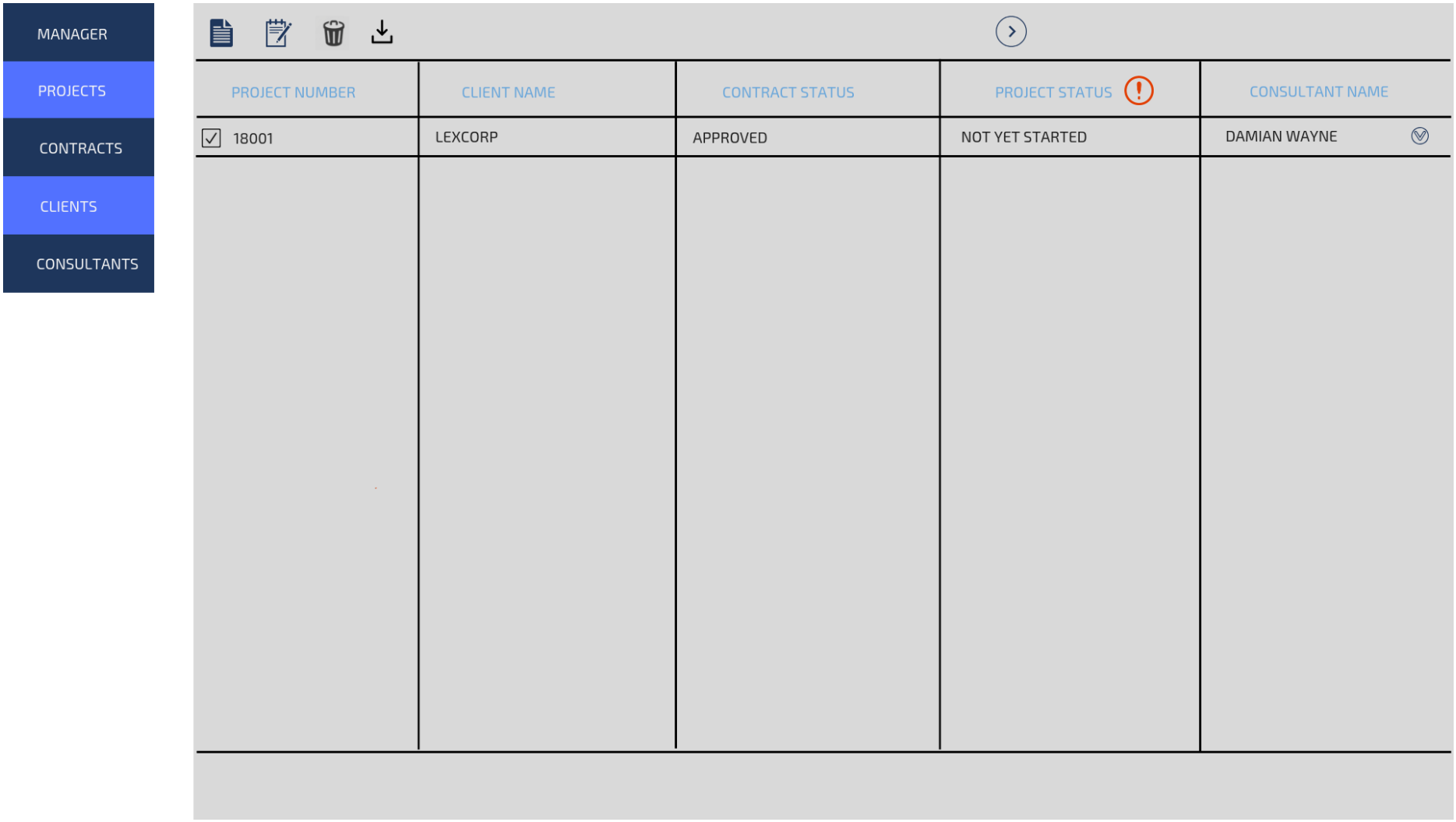
SCREEN 20

Functional Req. ID:	Functional Req. ID 12.



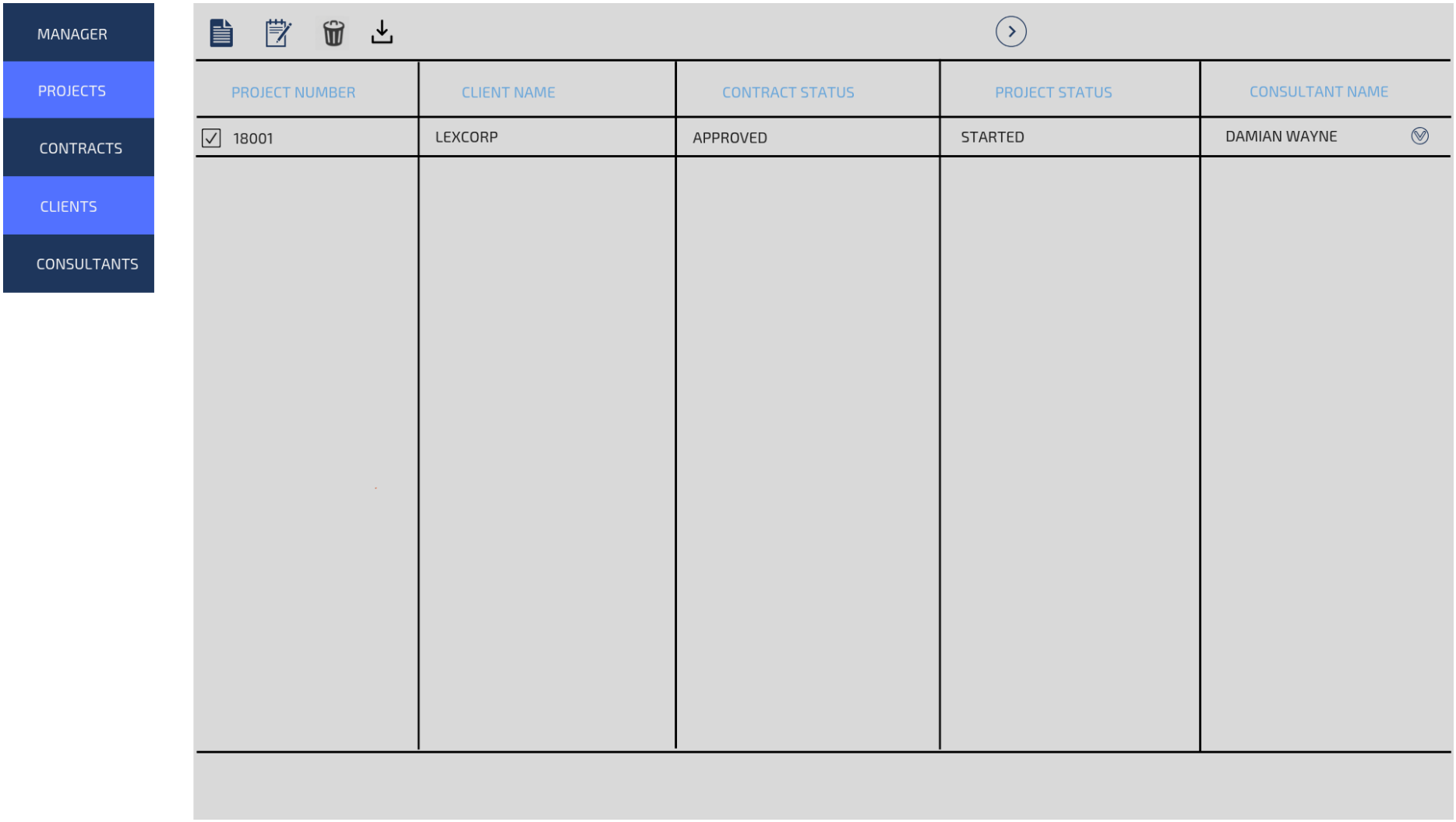
SCREEN 21

Functional Req. ID:	Functional Req. ID 13.



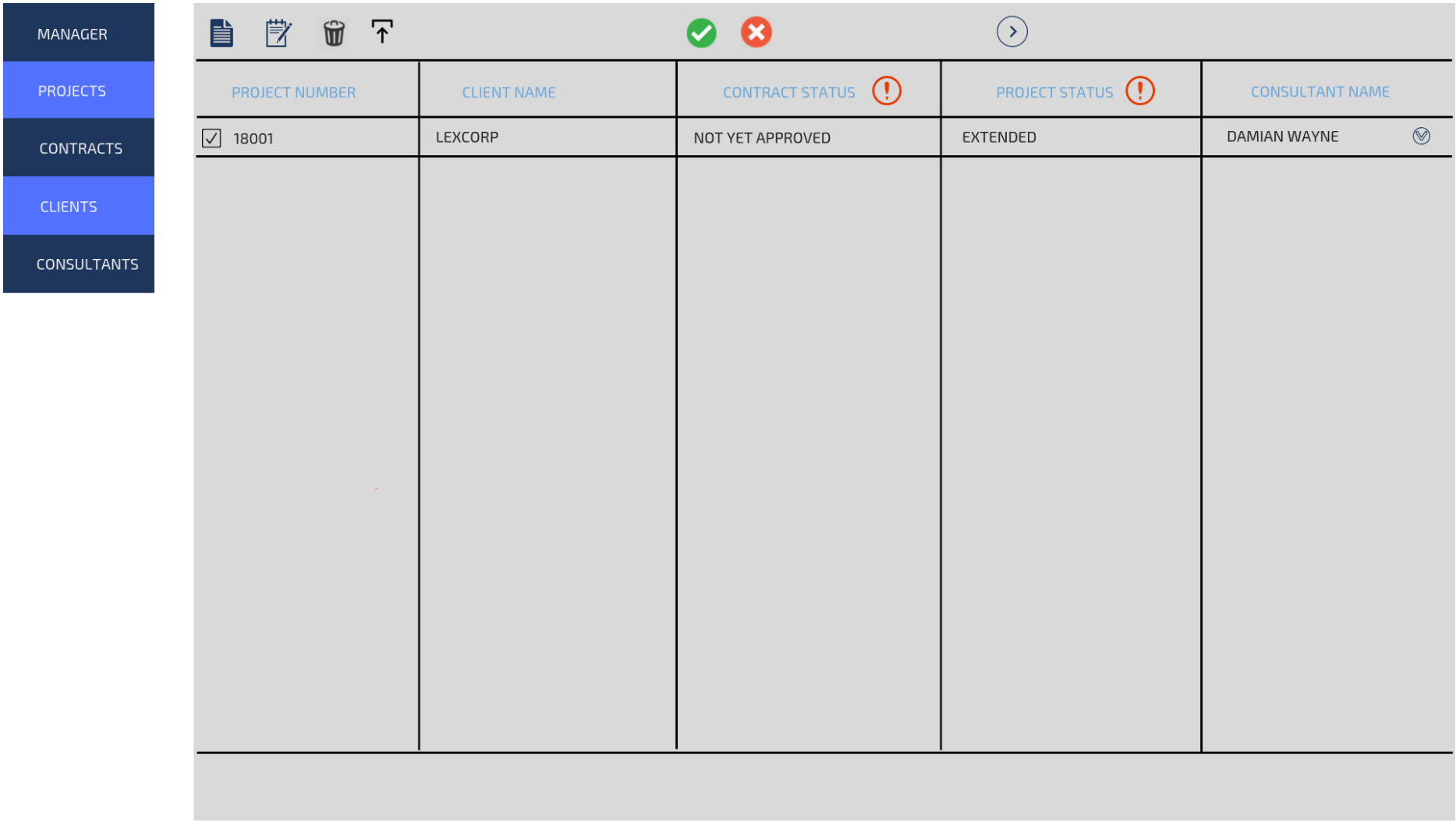
SCREEN 22

Functional Req. ID:	Functional req. ID 13.



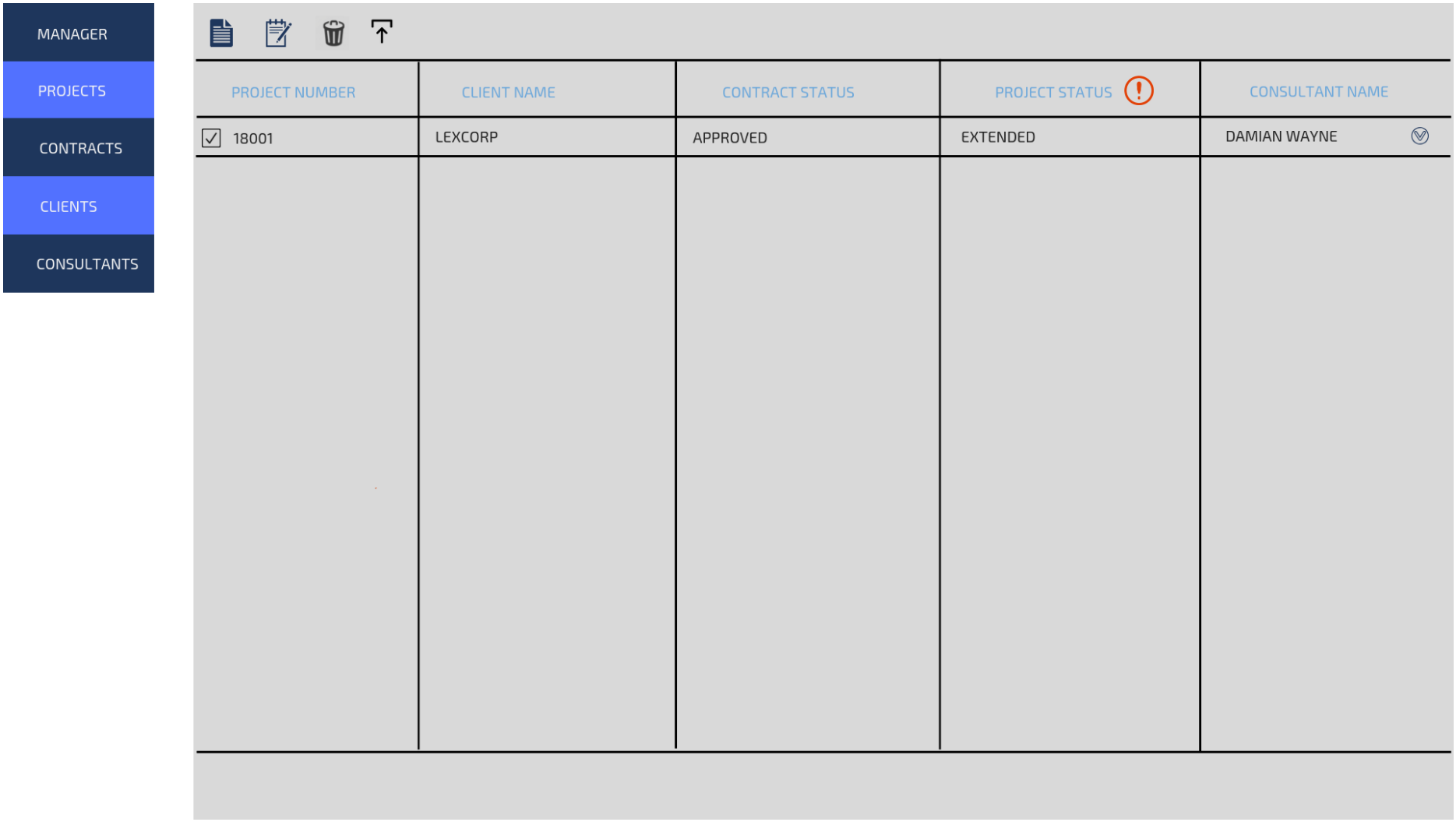
SCREEN 23

Functional Req. ID:	Functional Req. ID 13., Functional Req. ID 14.



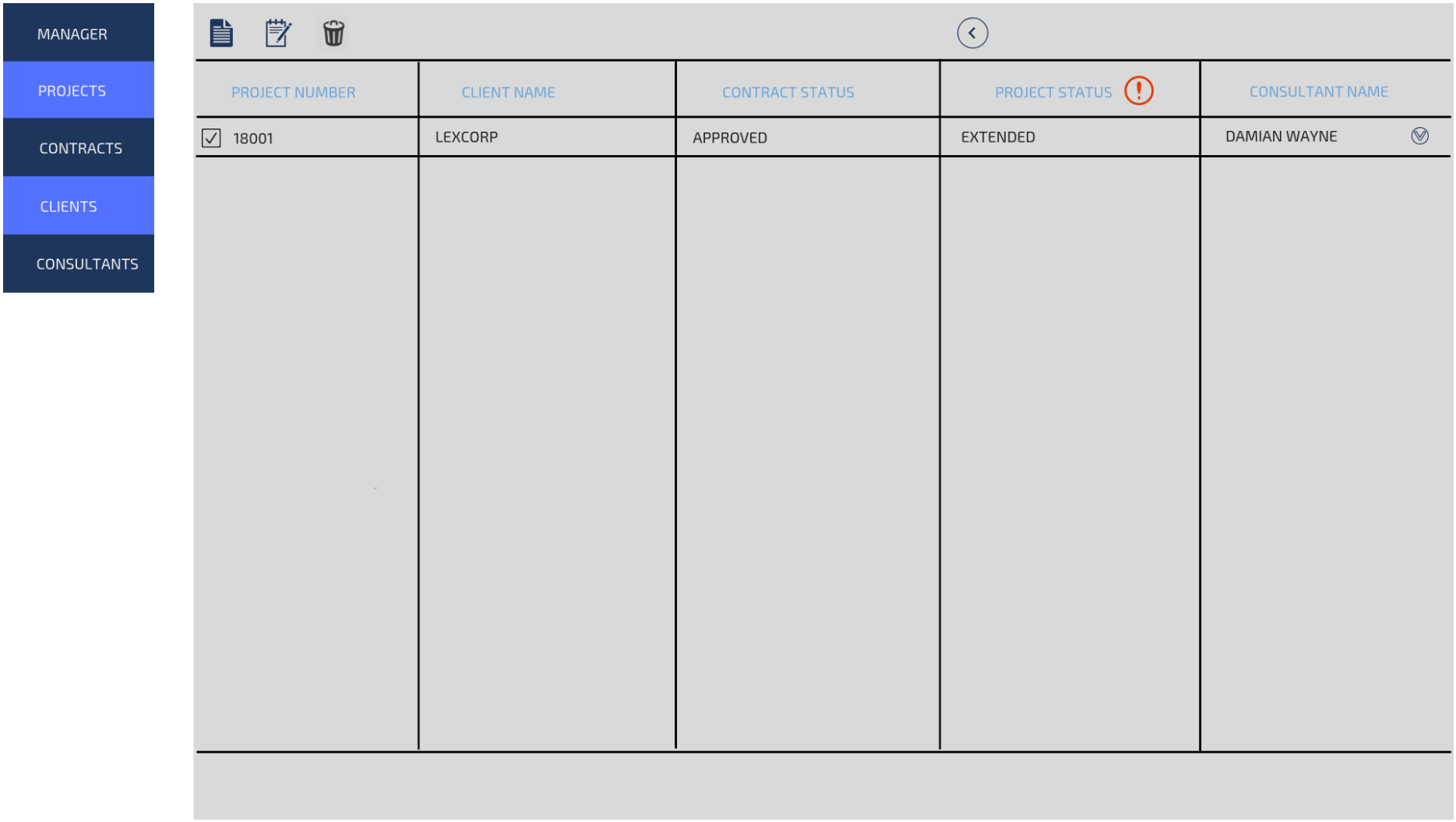
SCREEN 24

Functional Req. ID:	Functional Req. ID 14.



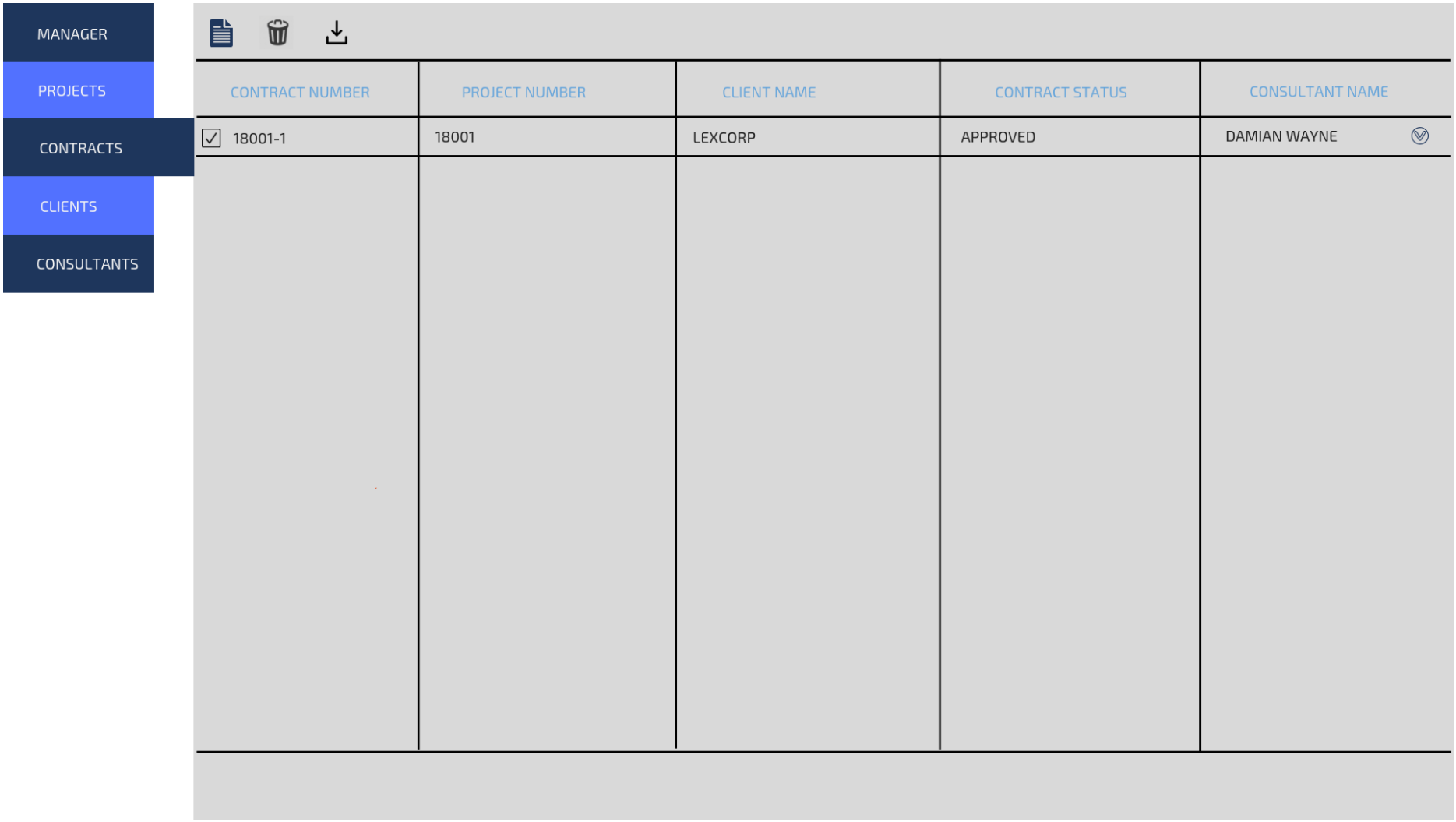
SCREEN 25

Functional Req. ID:	Functional req. ID 14.



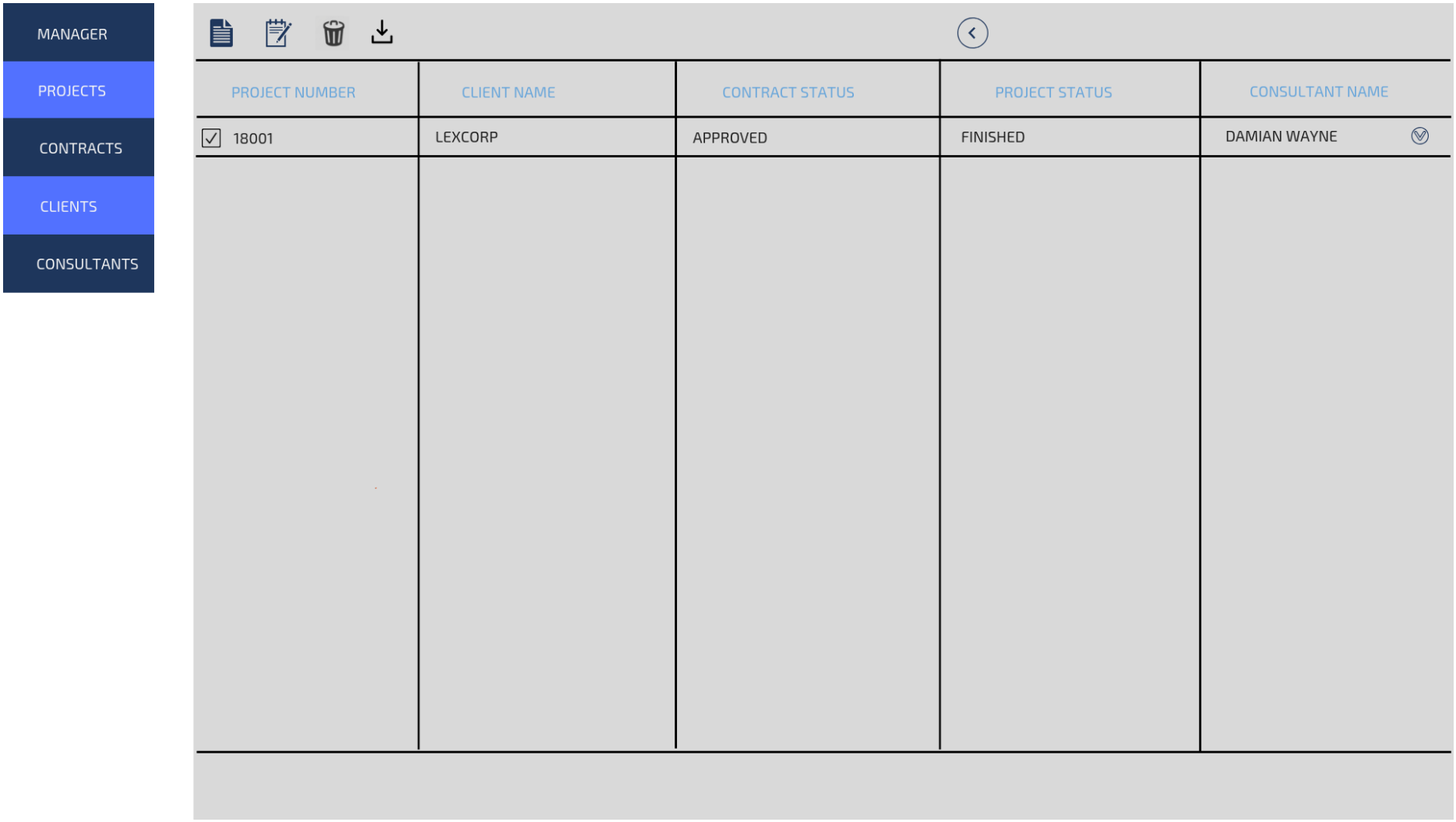
SCREEN 26

Functional Req. ID:	Functional Req. ID 15.



SCREEN 27

Functional Req. ID:	Functional Req. ID 13.



7. Tools Used

Tool used to create Database Model is:

Visual Paradigm Community edition
It was chosen for creating the physical ERD because of the following reasons:

- Familiarity of its environment.
- Desktop application where ERD models created can be generated into databases in MySQL format.
- Auto generate foreign keys by adding relationships between entities.
- Easy to resize, shape and move the entities and relationships in the diagram.
- Easy to update, add and remove columns.

Tools used to create the non- functional prototype of Web-Application Wireframe are:

Canva.com
It was chosen for this step because of the following reasons:

- Free-to use.
- Free to download pages in .jpeg or .png formats.
- Ability to customize text, shapes, colours, lines and dimensions of each page with a range of suitable options.
- Highly intuitive user-interface.
- Steep learning curve (relatively easy to learn within a short time-period).
- Provides a simple mechanism for layering picture components.

Invisionapp.com
It was chosen for this step because of the following reasons:

- The feature to upload mock-up images as separate files.
- Ability to connect mock-ups by clicking.
- Ability to select any position within the images for clicking.