Design Document - DD

January 2021

POLITECNICO
MILANO 1863

KONG XIANGYI
&
ZHANG YUEDONG

Contents

Chapter 1

INTRODUCTION

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Click Customer : The customer has the required technology to access the store. I.e a smartphone. They can use the customer terminal software.
- Brick Customer : The customer doesn't have the required technology to access the store, they have to hand out "tickets" on the spot.
- Store Manager : They have to manage the Store System, include the software and hardware.
- Ticket: The ticket is a document which contains three key information: QR Code, the estimated departure time, the queue number, and the Store Planned Roadmap. To the click customer, it's **E-ticket** but to the brick customer,it's **Paper Ticket**,and doesn't contain the estimated departure time, and just a General Store Map without the Planned Road.
- QR Code : When customer booked a visit, they will received a QR Code.

- QR Code Scanned Machine : A hardware, the Click Customer can use this machine scan their QR code.
- Tickets Hand-Out Machine : A hardware, the Brick Customer can use it retrieve their Ticket.
- Store Planned Roadmap: A store map that includes a finer way which is recommended form Store System.
- Digital Counterpart : A hardware, it with show the queue number.
- Store Back-End System : A software, as the back-end manages all stuffs.
- On-Time Store Data : A dataset that includes the store's on-time date.
  - The current queue
  - The customers in the store
  - The maximum number of people in the store.
- Long-Term Customers: The Click Customer who visited the store more than one time by the CLup mobile application.

1.3.2 Acronyms

- RASD - Requirement Analysis and Specification Document
- DD - Design Document
- CLup - Customers Line-up
- UI - User Interface
- IOS - iPhone OS
- PC - Personal Computer
- IaaS - Infrastructure as a Service
- CRM - Customer Relationship Management
- LAN - Local Area Network

1.3.3 Abbreviations

-

1.4 Revision history

1.5 Document Structure

Chapter 2

ARCHITECTURAL DESIGN

In this chapter, we will describe the architectural design of our system.

We will use the Top-down design approach, design the very high-level structure first, and then gradually work down to detailed decisions about low-level constructs. Finally, arrive at detailed decisions.[4] Let us start with the High-level components and their interaction.

2.1 Overview: High-level components and their interaction

We chose **3-Tiered architecture** with the **Thin Client** strategy for our system. As shown in Fig.2.1.[1]

Tier-1 is the presentation layer. This layer will deploy the Click Client's mobile application, Store Manager's Management Web Page, and even Digital Counterpart and Ticket Hand-Out Machine's presentation.

Tier-2 is the logic application layer. This layer will deploy our Back-End System component.

Tier-3 is the data access layer. This layer includes our DBMS and the Data Base.

We have a web page as the Management page for the Store Manager but considered this page a static page, and the traffic will be minimal, so we did not use the 5-Tiered Architecture with the Web Server and the Script Engine Server to general the Dynamic Page.

Finally, our high-level architecture is shown in Fig.2.2. The Store Manager's PC and other hardware will connect with an Ethernet cable. The Click Customer's Mobile App communicates with our Back-End System with the Internet.

More detailed components will be introduced in the next section.

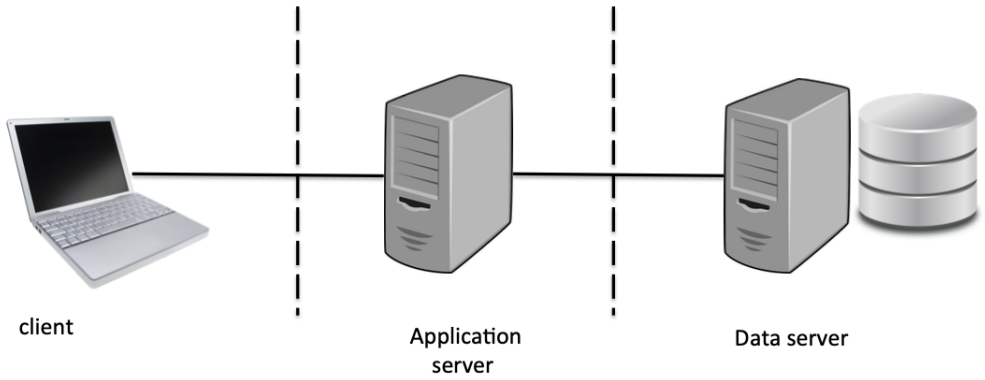

Figure 2.1: 3-Tiered architecture

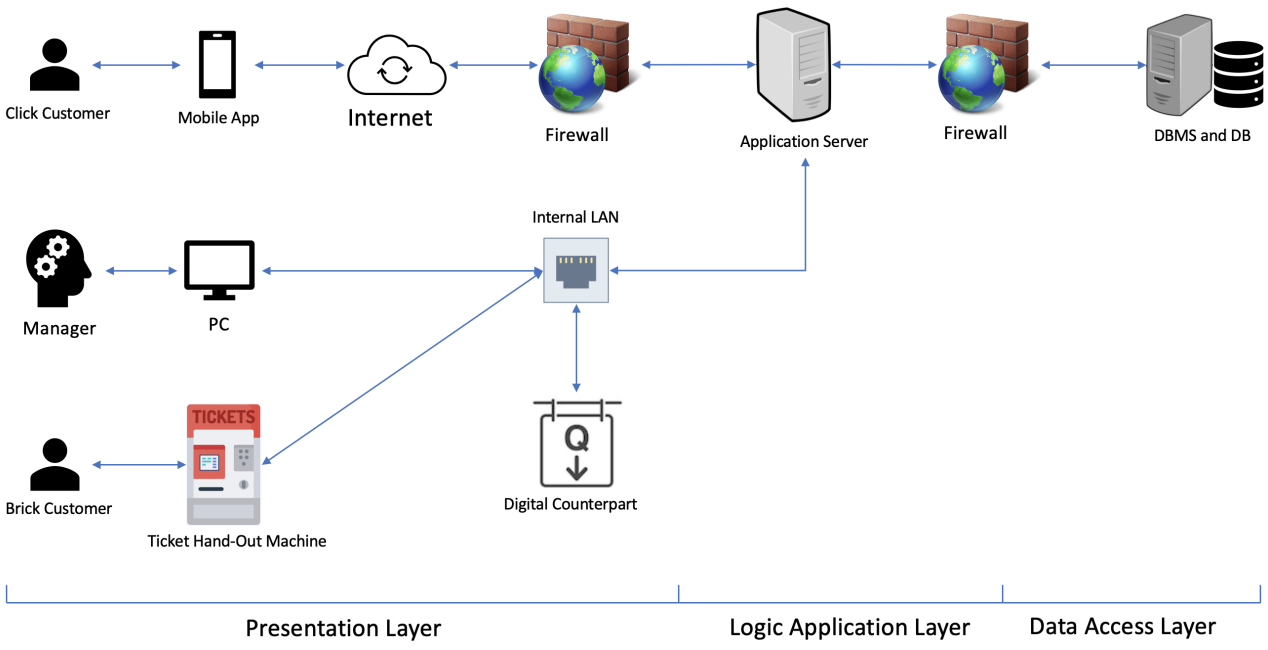

Figure 2.2: High level architecture

2.2 Component View

The following diagram Fig.2.3 is the **Component Diagram**, and it contains all components in our system. This diagram shows the 3-Tiered Architecture of our system. The Yellow components are in Presentation Layer. The blue components are in the Logic Application Layer, and the Green components DBMSServer is in the Data Access Layer. We will separately introduce all these components in this section and introduce all interfaces at Sec.2.5.

The **Redirector** component is a bridge for communicating the Presentation Layer and the Logic Application Layer. It is transparent to User and back-end systems, plays the role of forwarding information on both sides, and connect interfaces. It has four subsystems, respectively, responsible for communicating with four types of devices, as shown in Fig.2.4.

The **ClickCustomerRedirector** will transfer the message between Mobile Application and the Application Server. The Register message will pass by the RegisterNewClickCustomer, and look out the Valid/History Booking will use the GetClickCustomerData Interface, this Interface will return all information of this Customer. For the most important operation - Booking a visit, use GetBookingSchedule to get the available Time/Date. Then, through the ModifyClickCustomerData Interface to submit the Booking, by the way, it also can modify other available data of this Customer. Moreover, the SendNotifyToClickCustomer Interface will handle Notify message. The Mobile Application can actively "pull" Notify message through this Interface from the Back-End System. To "push" Notify information, we will use the Observer Pattern. It will introduce in Sec.2.6

The **TicketMachineRedirector** will transfer the Ticket Hand-Out Machine's message. It needs two pieces of information, the allowed maximum number of people in the store and how many Customers in the Queue in time. So The GetQueueSchedule and GetMaxNumberInStore Interface can do these operations, and these two fields will show on the Ticket Machine's screen. Finally, the retrieve ticket operation will handle by the ModifyQueueSchedule Interface. It will put a BrickCustomer into the Queue.

The **DigitalCounterpartRedirector** will help the Digital Counterpart display the next queue number. The Customer can enter the store after found his number on the Digital Counterpart.

The **ManagementSystemRedirector** will function for the Management System of our Store Manager. This redirector will transmit the message between the Presentation layer's web page and the Back-End System. Through the Interface shown in Fig.2.5, the Manager can realize his operation like reschedule someone's Booking, adjust the Queue order, or lower the maximum number.

The next three subsystems are the core subsystems in our application layer.

2.3 Deployment View

2.4 Runtime View

2.5 Component Interfaces

2.6 Selected Architectural Styles and Patterns

2.7 Other Design Decisions

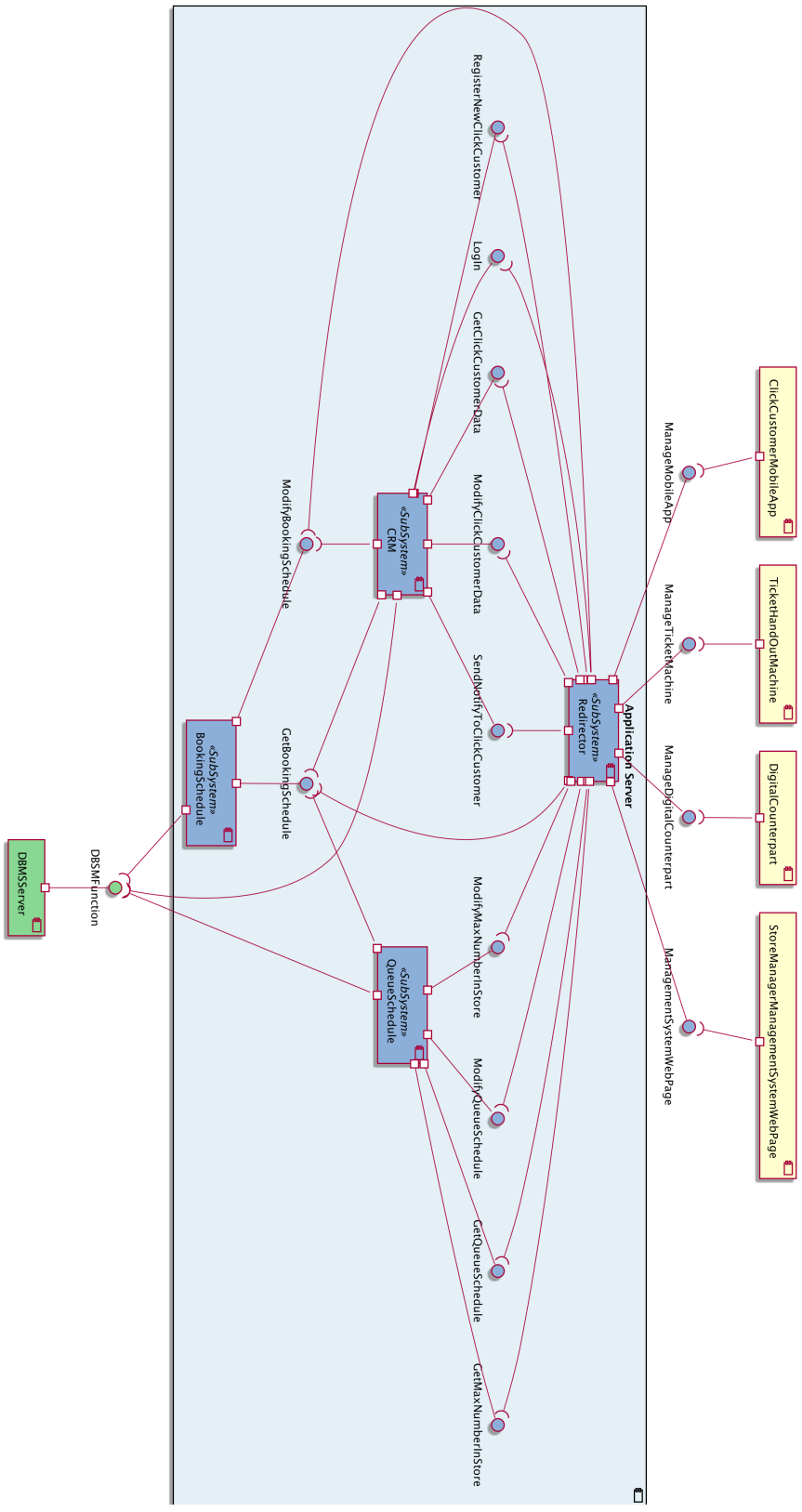

Figure 2.3: Component Diagram

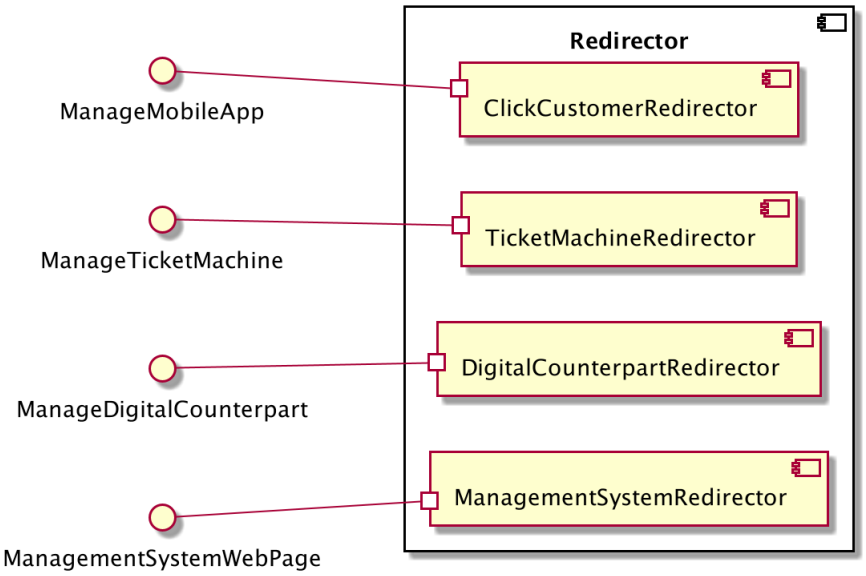

Figure 2.4: Redirector Component Diagram

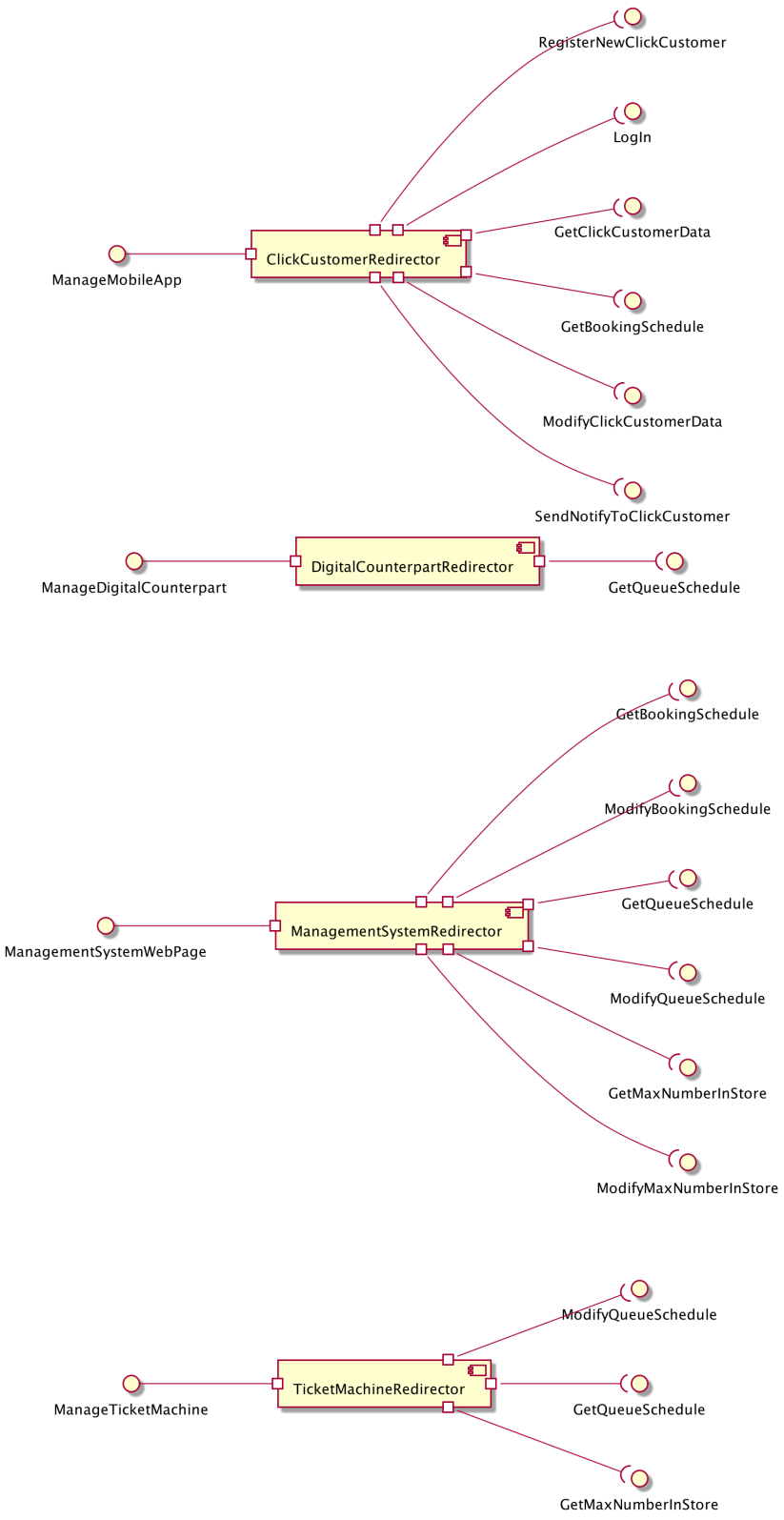

Figure 2.5: Redirector's Subsystem Component Diagram

Chapter 3

USER INTERFACE DESIGN

Chapter 4

REQUIREMENTS TRACEABILITY

Chapter 5

IMPLEMENTATION, INTEGRATION AND TEST PLAN

Chapter 6

Effort Spent

- **Kong XiangYi**

Date	Task	Hours
2021/01/02	Group discussion and task assignment	2h

- **Zhang YueDong**

Date	Task	Hours
2021/01/01	Launch DD	2h
2021/01/02	Group discussion and task assignment	2h
2021/01/03	Did S.2.2.1 and S.2.2.2 Component Diagram	3h
2021/01/04	Did S.2.2. Component Diagram describe	2h

Bibliography

[1] Cinzia Cappiello, Mariagrazia Fugini, Paul Grefen, Barbara Pernici, Pierluigi Plebani, Monica Vitali. (7 settembre 2018) Fondamenti di Sistemi informativi: per il Settore dell'Informazione.

[2] "IEEE Standard for Information Technology–Systems Design–Software Design Descriptions," in IEEE STD 1016-2009 , vol., no., pp.1-35, 20 July 2009, doi: 10.1109/IEEESTD.2009.5167255.

[3] "ISO/IEC/IEEE Systems and software engineering – Architecture description," in ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000) , vol., no., pp.1-46, 1 Dec. 2011, doi: 10.1109/IEEESTD.2011.6129467.

[4] Elisabetta Di Nitto, Matteo Rossi. (A.Y.2020/2021) The slides of the Software Engineering 2 course.