Université Concordia University

GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

SOEN 6461: SOFTWARE DESIGN METHODOLOGIES

COURSE PROJECT - WINTER 2023

Deliverable 2

PROJECT REPORT

iGO

Submitted To: Prof. Pankaj Kamthan

By Team D,

Yashwanth Gundlapally (40164633)
Apekshaba Gohil (40203058)
Pratik Gondaliya (40194062)
Carlos Garcia (40220038)
Ernesto Garsiya Melikhov (40039957)
Amro Elbahrawy (40221760)

Github:
https://github.com/mrgps1999/SOEN6461_Ticket_Vending_Machine-TVM-

Contents

Problem 6

6.1 High-level Solution Domain Model

Since the project's design follows Object-Oriented principles, it was decided that the solution domain model for iGo should follow Responsibility-Driven Design (RDD) principles. Based on that, the model was presented in the form of a Class-Responsibility-Collaborator (CRC) Card model.
The Use Case Diagram conducted in **Deliverable 1** was used as reference, where each use case was analyzed in terms of possible scenarios. One or more classes, along with their responsibilities and collaborator classes, were considered for each use case. Related classes were linked using directed lines, resulting in a cyclic directed CRC model that acts as a high-level solution domain model for the system. This will be used as a reference to implement a fully-structured Class Diagram.
The main class is **UI**, which acts as a controller that processes user input commands to the models and database. Other classes can be categorized into:

- **Billing/Finance**: Responsible for handling payment and transaction processes. The included classes are **TransactionController, Payment, Transaction, Invoice, PaymentPartner**.
- **Users**: The users of the system. Namely **Conductor and Commuter**. Commuter includes both registered and unregistered commuters.
- **Business**: Which includes the business logic of the system, including tickets, fares, rechargeable cards, and TVM. Classes are **iGoCard, Ticket,Fare, TVM**.

The following figure presents the CRC Card model for iGo:

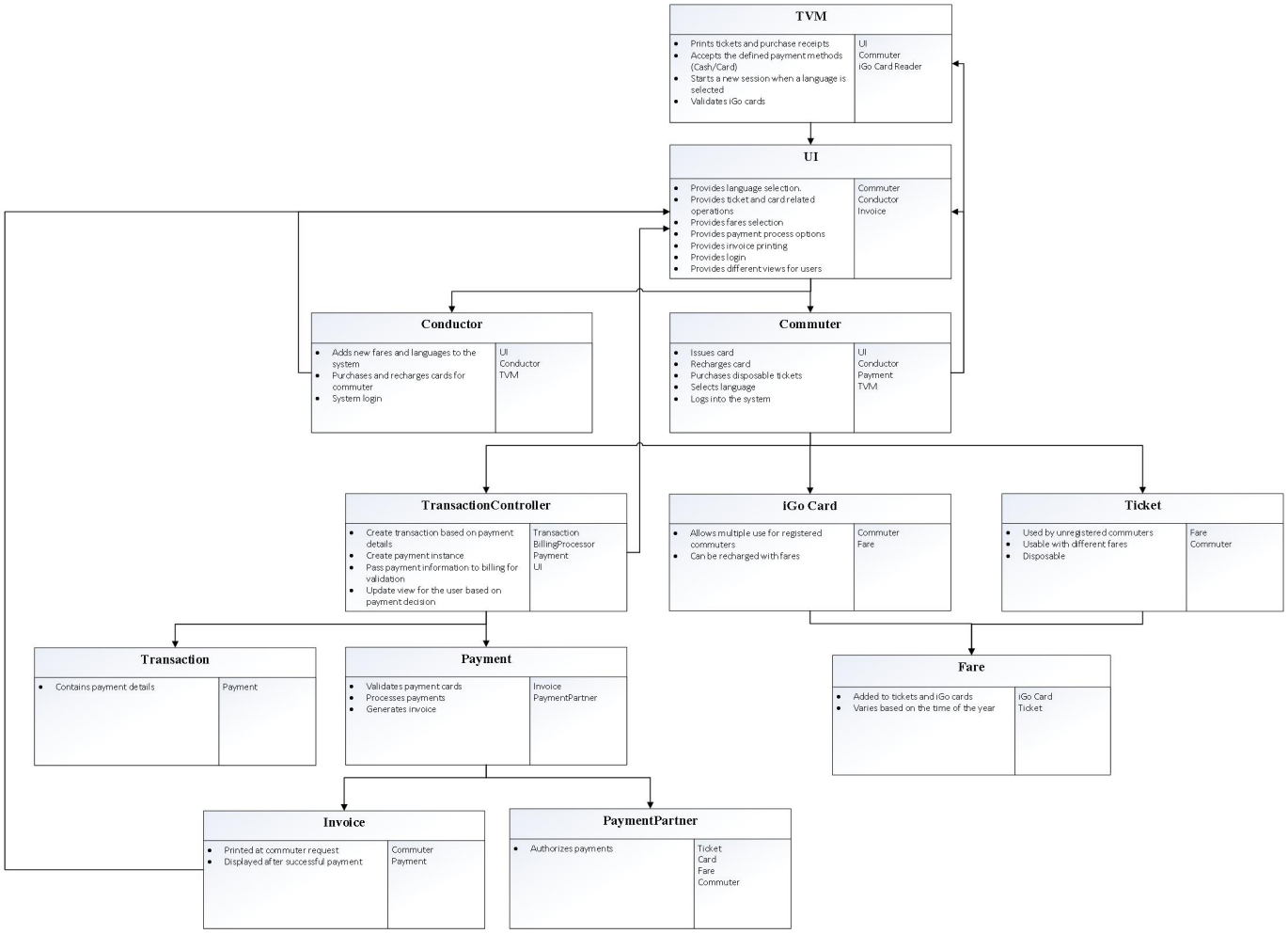

Figure 6.1: CRC Card Model

Problem 7

7.1 Low-level Solution Domain Model- Class Diagram

7.1.1 iGO Class Diagram

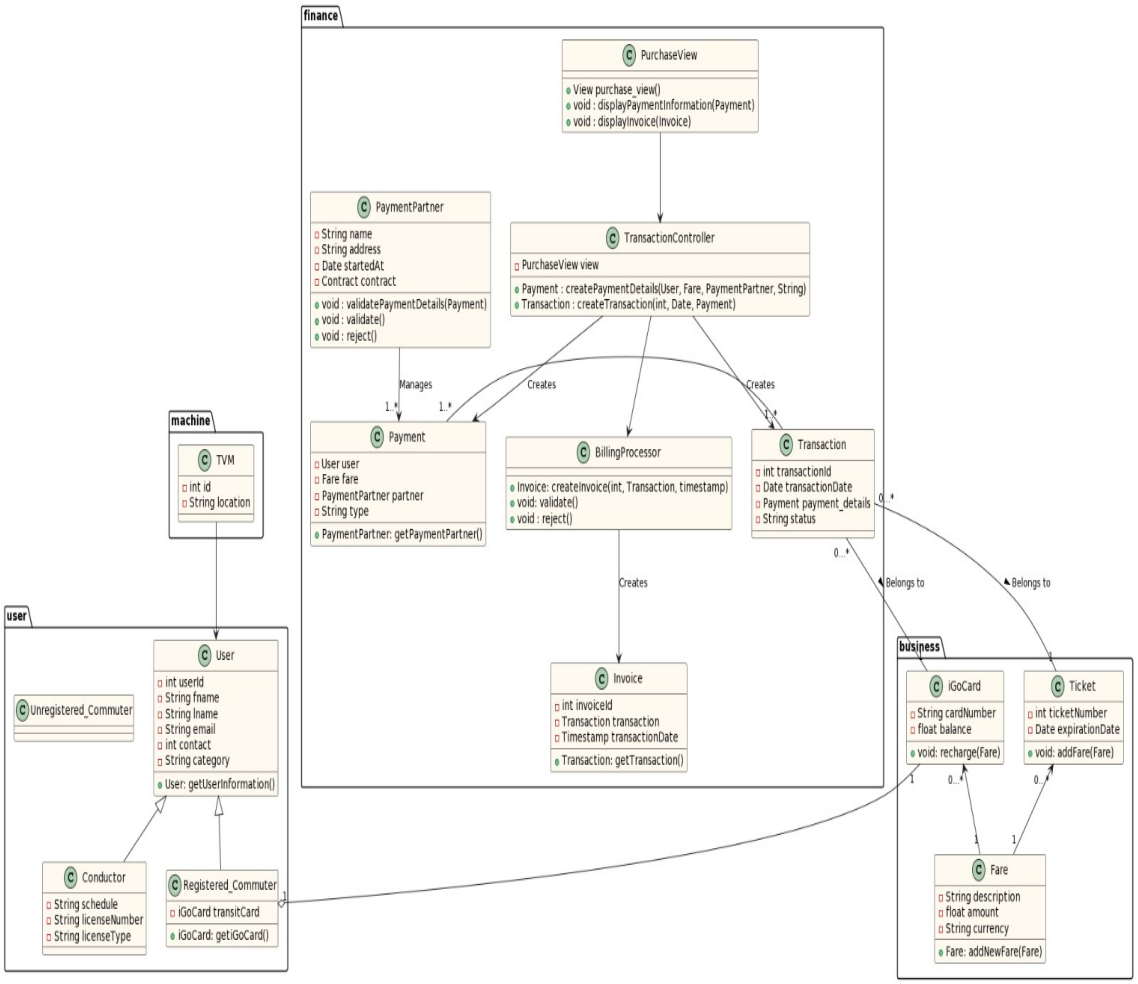

Figure 7.1: iGO Class Diagram

7.2 Low-level Solution Domain Model - Sequence Diagrams

7.2.1 iGO Sequence Diagram

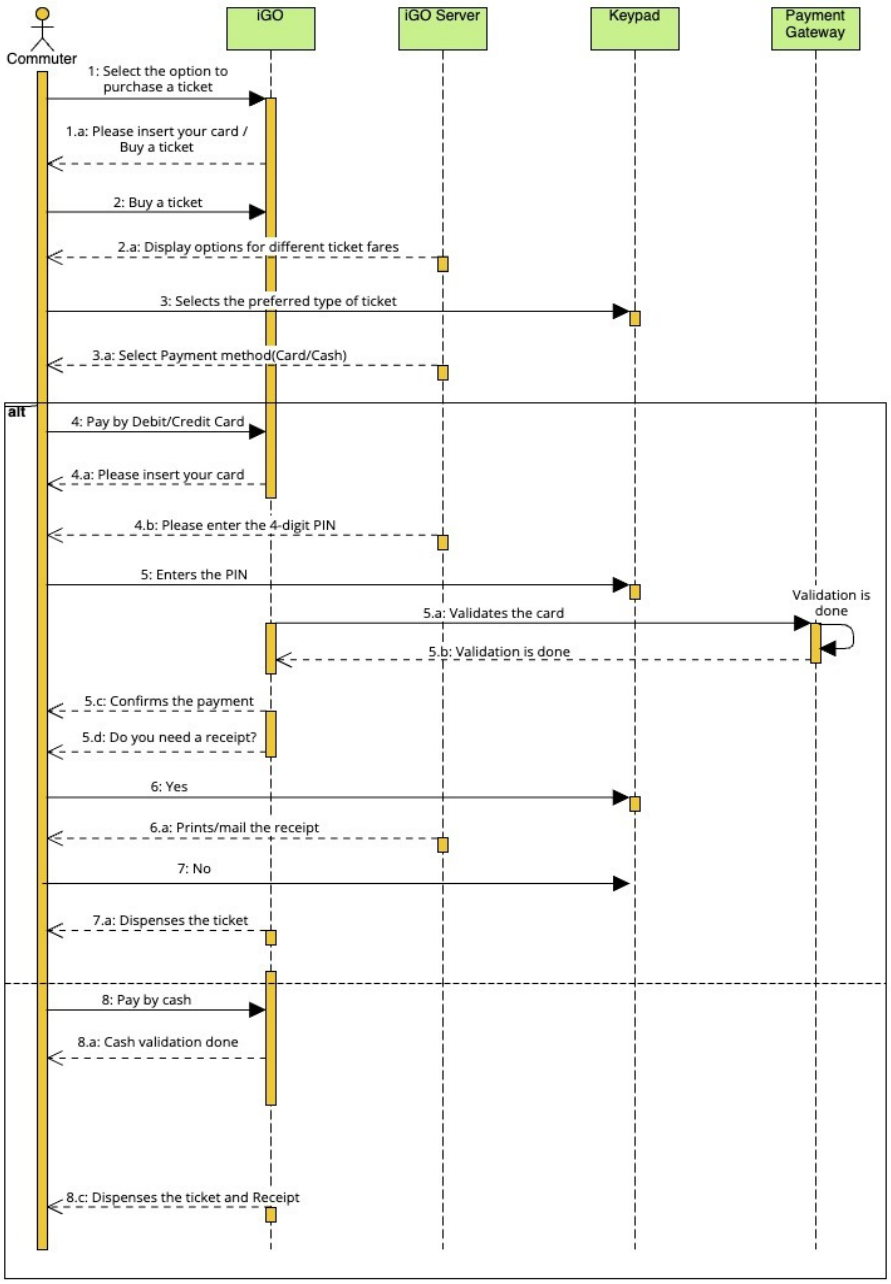

Figure 7.2: iGO Sequence Diagram

7.2.2 Payment using card Sequence Diagram

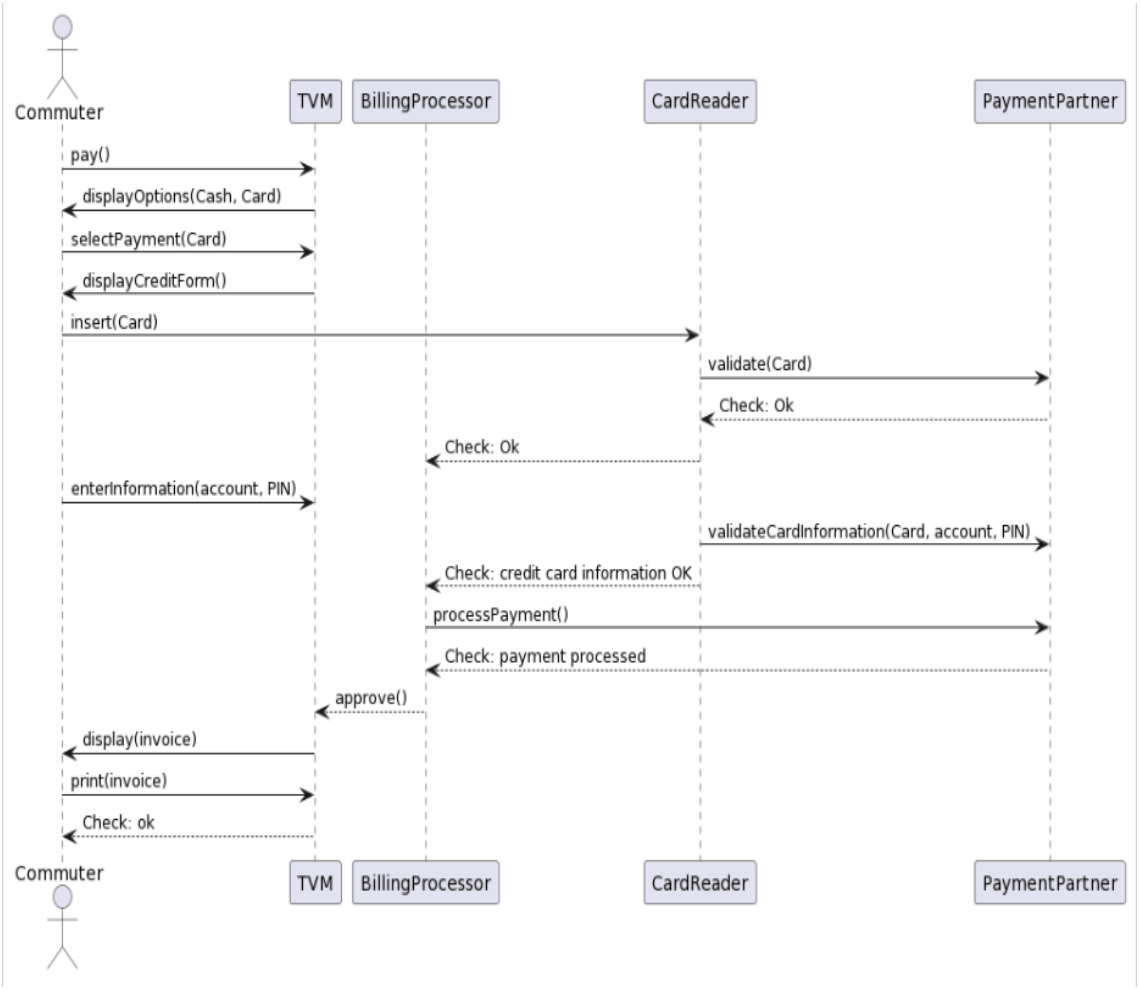

Figure 7.3: Payment using card Sequence Diagram

7.2.3 Make Payment in Cash

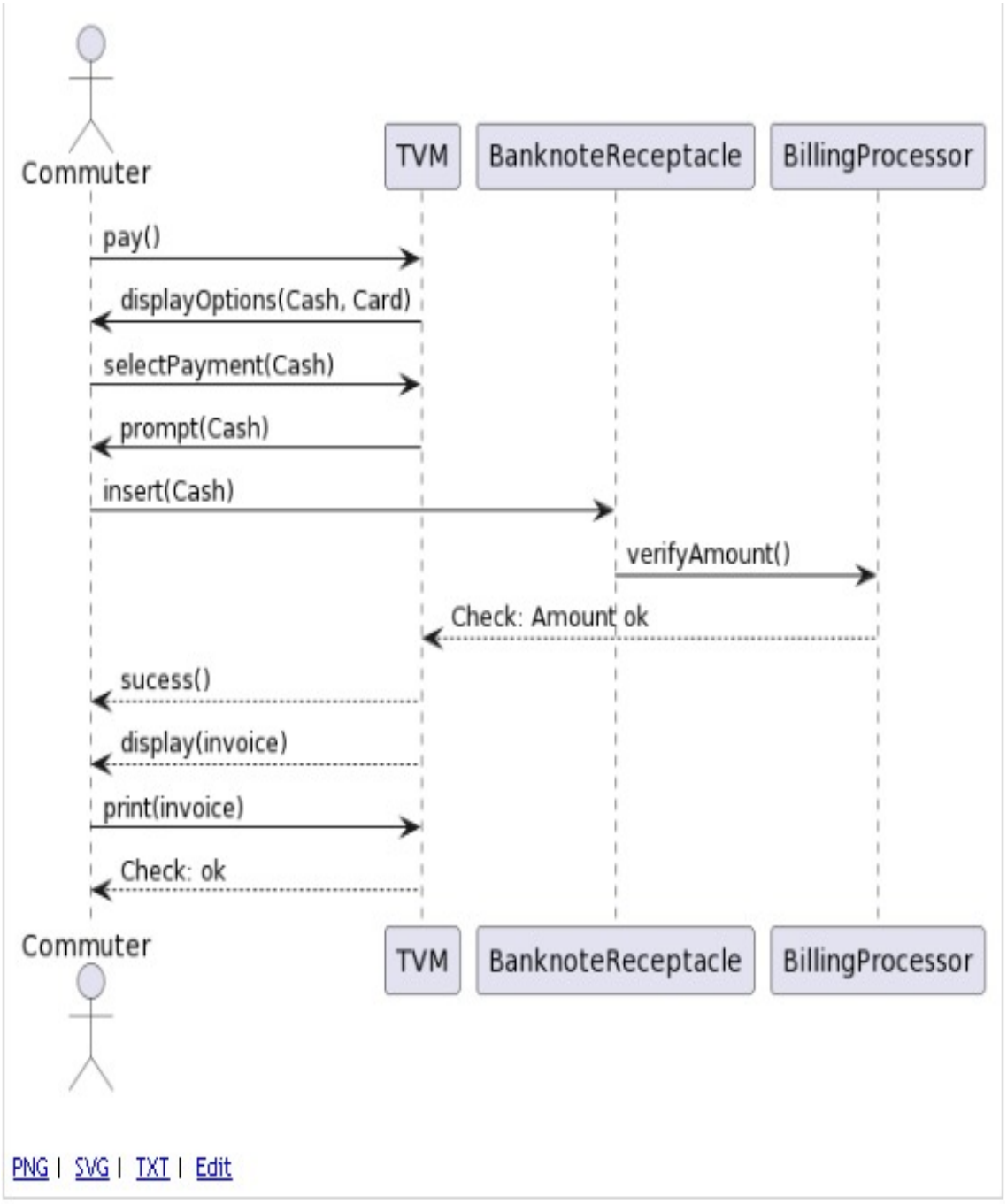

Figure 7.4: Payment in cash Sequence Diagram

7.2.4 Recharge Card

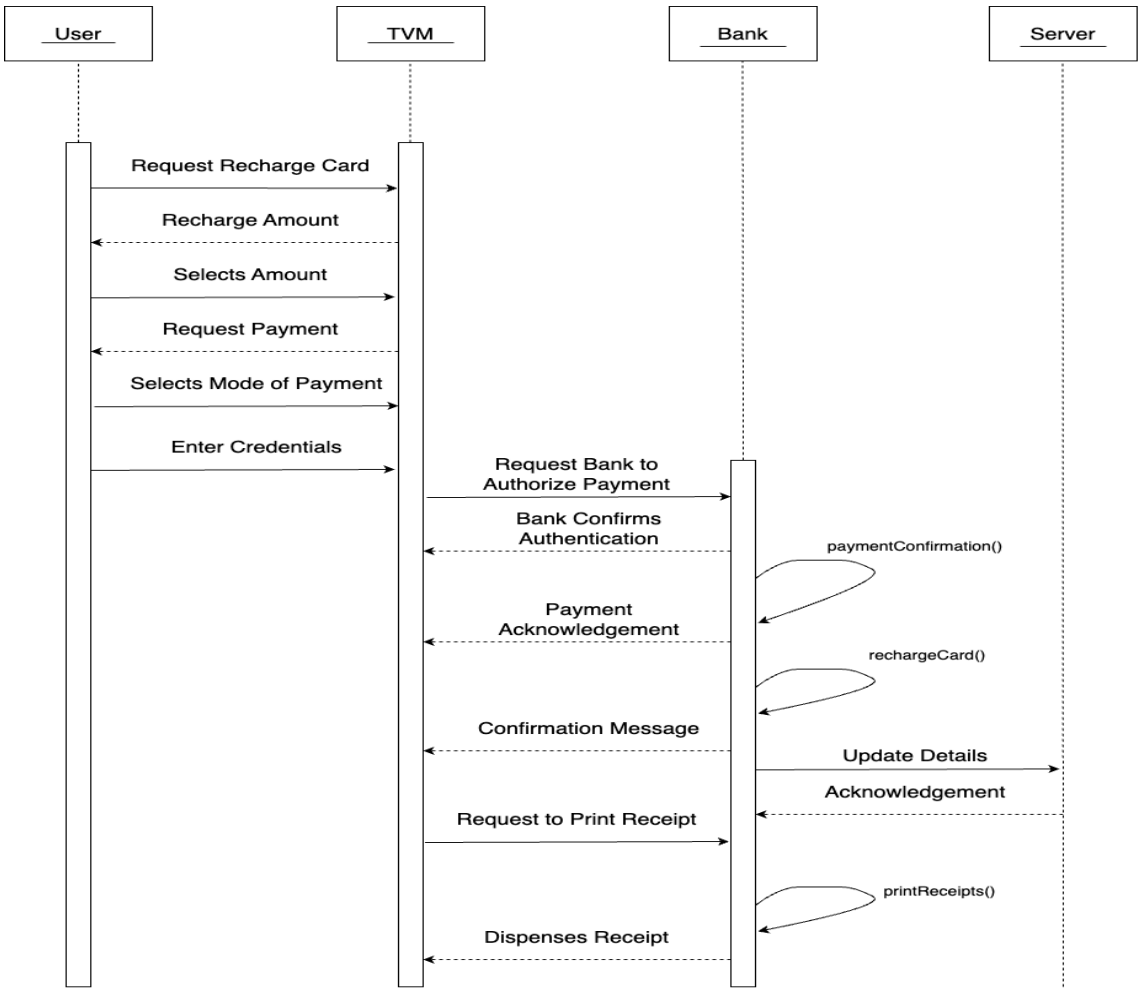

Figure 7.5: Recharge Card Sequence Diagram

7.2.5 Recharge Card

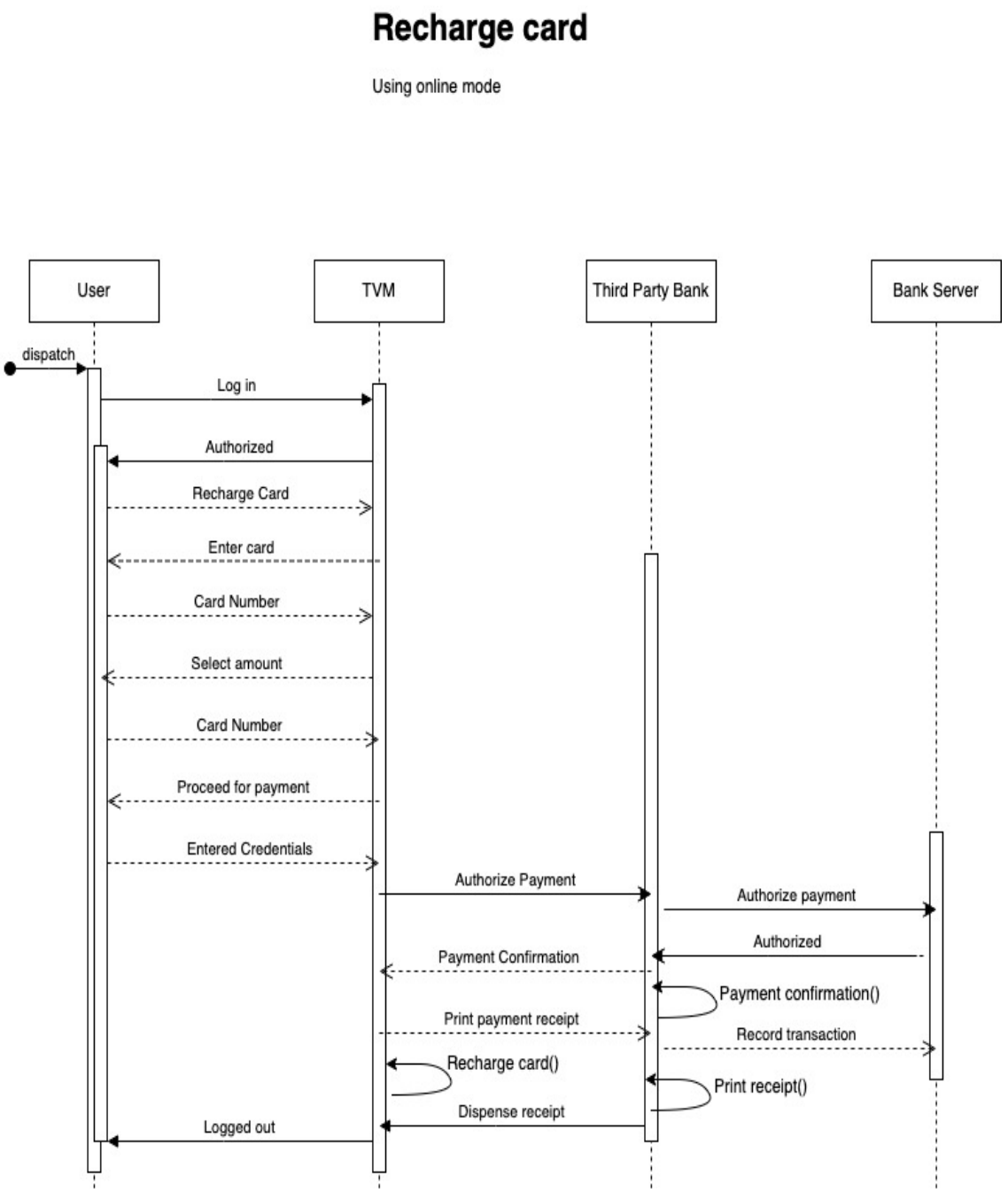

Figure 7.6: Recharge Card Using Online mode Sequence Diagram

7.2.6 Purchase Ticket

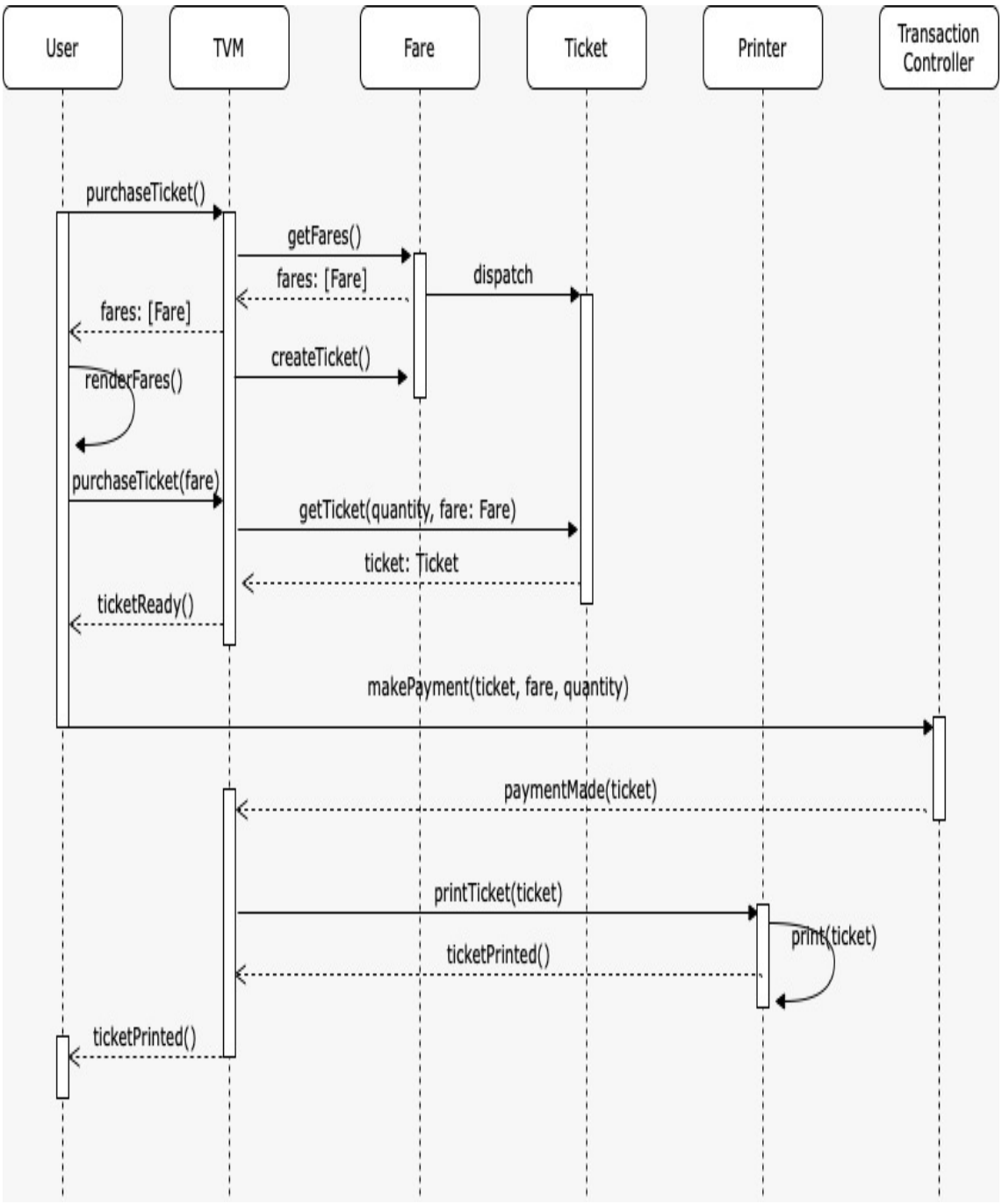

Figure 7.7: Purchase Ticket Sequence Diagram

Problem 8

8.1 Python Code

1. In iGo implementation user can be of two types one is Child or Senior Citizens and another one is adult based on types of user fare will be decided. For demonstration purpose we gave access of iGo system to two types of users which are adult and child.

2. **Select Language screen:** This is the first screen a user will see as soon as all things go as per the requirement of the iGo system, then the user will have two options in front of them which are English and French "Language options", Since a large group of English and French-speaking people are expected to be the potential users. two languages(English and French) are displayed to choose between:

```
def select_language_display(username):
    print("Display Select Langauge:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=350,height=350,bg="white")
    frame.place(x=480,y=70)
    heading = Label(frame, text='Select Language', fg='#57a1f8',bg=
   'white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=80,y=5)
    Button(frame,width=25,pady=7,text="English",bg='#ff8c00',fg='
   #57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),
   border=0,highlightbackground="#ff8c00",command=lambda:
   English_Language(username)).place(x=25,y=107)
    Button(frame,width=25,pady=7,text="French",bg='#ff8c00',fg='#57
   a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),
   border=0,highlightbackground="#ff8c00",command=lambda:
   French_Language(username)).place(x=30,y=170)
```

3. **iGO Menu:**After selecting English or French user will have two options which are Recharge card and Buy Ticket.

   After selecting Recharge Card or Buy Ticket user will have a set of options in iGo. These are Monthly Passes, Weekly Passes, Day Pass/Ticket, Weekend Pass, and One-way tickets. If the user will select French, then all options from now onwards will be in French only.

```
def Select_Plans(username):
    print("Display Select Plans:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=450,y=70)
    heading = Label(frame, text='Select Language', fg='#57a1f8',bg='white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=70,y=5)
    Button(frame,width=25,pady=7,text="Monthly Pass",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: Monthly_pass(username)).place(x=25,y=107)
    Button(frame,width=25,pady=7,text="Weekly Pass",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: Weekly_pass(username)).place(x=30,y=170)
    Button(frame,width=25,pady=7,text="Day Pass",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: Day_pass(username)).place(x=35,y=233)
    Button(frame,width=25,pady=7,text="Weekend Pass",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: Weekend_pass(username)).place(x=40,y=296)
    Button(frame,width=25,pady=7,text="One Way Ticket",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: One_way_pass(username)).place(x=45,y=359)
```

- After clicking on any of the above options user will have specific iGo layout based on the option which user is selecting. Apart from, Day Pass/Ticket and One way Ticket option user will have button of make payment directly with the price which user needs to pay for selecting any of the options from Monthly, Weekly or Weekend individually as per user's selection. But when it comes to Day Pass/Ticket and one One way ticket then user needs to enter no of tickets or pass which he needs. So, here it's possible that user might enter nothing and try to press button named "Procced for Payment" in this case user will have popup showing message called "invalid input". If user click "ok" button which will be there in popup then user will be redirected to selection options display where user will have again all the option which has been discussed earlier. Here we have handled exception as well so iGo might not get crashed.

4. **Monthly Pass:** The below method will display the details of the monthly pass:

```
def Monthly_pass(username):
    print("Display Monthly Pass:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=480,y=70)
    heading = Label(frame, text='Monthly Pass', fg='#57a1f8',bg='white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
```

```
    if username == 'child' or username == 'senior citizens':
        button = Button(frame,width=25,pady=7,text="Make Payment:
    $50",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
    ,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
    command=lambda: select_payment_option(username)).place(x=35,y
    =120)
    else:
        button = Button(frame,width=25,pady=7,text="Make Payment:
    $60",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
    ,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
    command=lambda: select_payment_option(username)).place(x=35,y
    =120)
```

5. **Weekly Pass:** The below method will display the details of the weekly pass:

```
def Weekly_pass(username):
    print("Display Weekly Pass:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=440,y=70)
    heading = Label(frame, text='Weekly Pass', fg='#57a1f8',bg='
    white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    if username == 'child' or username == 'senior citizens':
        button = Button(frame,width=25,pady=7,text="Make Payment:
    $10",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
    ,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
    command=lambda: select_payment_option(username)).place(x=35,y
    =120)
    else:
        button = Button(frame,width=25,pady=7,text="Make Payment:
    $15",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
    ,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
    command=lambda: select_payment_option(username)).place(x=35,y
    =120)
```

6. **Day Pass:** The below method will display the details of the Day pass:

```
def Day_pass(username):
    print("Display Day Pass:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=480,y=70)
    heading = Label(frame, text='Day Pass', fg='#57a1f8',bg='white'
    ,font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    No_ticket = Label(frame, text = "No. of Tickets",fg="#57a1f8",
    font=('Microsoft YaHei UI Light',15,'bold'))
    No_ticket.place(x = 30,y = 70)
    tickets = Entry(frame, width=25,fg='black',border=0,bg="white",
    font=('Microsoft YaHei UI Light',11))
    tickets.place(x=150,y=73)
    tickets.insert(0,'')
    tickets.bind('<FocusIn>',on_enter)
```

```
    tickets.bind('<FocusOut>',on_leave)
    Frame(frame,width=200,height=1.5,bg='black').place(x=140,y=100)
    button = Button(frame,width=25,pady=7,text="Proceed For Payment
",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,
'bold','italic'),border=0,highlightbackground="#ff8c00",command=
lambda: tickets_calculation(tickets,username)).place(x=35,y=120)
```

7. **Weekend Pass:** The below method will display the details of the Weekend pass

```
def Weekend_pass(username):
    print("Display Weekend Pass:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=440,y=70)
    heading = Label(frame, text='Weekend Pass', fg='#57a1f8',bg='
white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    if username == 'child' or username == 'senior citizens':
        button = Button(frame,width=25,pady=7,text="Make Payment:
$4",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
command=lambda: select_payment_option(username)).place(x=35,y
=120)
    else:
        button = Button(frame,width=25,pady=7,text="Make Payment:
$5",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light'
,15,'bold','italic'),border=0,highlightbackground="#ff8c00",
command=lambda: select_payment_option(username)).place(x=35,y
=120)
```

8. **One Way Ticket:** The below method will display the details of one-way ticket:

```
def One_way_pass(username):
    print("Display One Day Pass:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=480,y=70)
    heading = Label(frame, text='One way Pass', fg='#57a1f8',bg='
white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    No_ticket = Label(frame, text = "No. of Tickets",fg="#57a1f8",
font=('Microsoft YaHei UI Light',15,'bold'))
    No_ticket.place(x = 30,y = 70)
    tickets = Entry(frame, width=25,fg='black',border=0,bg="white",
font=('Microsoft YaHei UI Light',11))
    tickets.place(x=150,y=73)
    tickets.insert(0,'')
    tickets.bind('<FocusIn>',on_enter)
    tickets.bind('<FocusOut>',on_leave)
    Frame(frame,width=200,height=1.5,bg='black').place(x=140,y=100)
    button = Button(frame,width=25,pady=7,text="Proceed For Payment
",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,
'bold','italic'),border=0,highlightbackground="#ff8c00",command=
lambda: tickets_calculation(tickets,username)).place(x=35,y=120)
```

- For the two options which are Day Pass/Ticket and One way ticket if user enter correct input and for the rest of the other option after selecting “Make Payment button user will have two options which are “Payment Options” named as “Credit card/ Debit card” and “Cash

9. **Checkout screen:** The below methods will calculate the price for the selected tickets/pass and then display on the screen

```
def tickets_calculation(tickets,username):
    print("Monthly Plan:"+username)
    no_tickets = tickets.get()
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=480,y=70)
    if username == 'child' or username == 'senior citizens':
        total_cost = float(no_tickets) * 1.99
        print(total_cost)
        button = Button(frame,width=25,pady=7,text="Make Payment of
    "+str(total_cost),bg='#ff8c00',fg='#57a1f8',font=('Microsoft
   YaHei UI Light',15,'bold','italic'),border=0,highlightbackground
   ="#ff8c00",command=lambda: select_payment_option(username)).
   place(x=35,y=120)
    else:
        total_cost = float(no_tickets) * 2.99
        button = Button(frame,width=25,pady=7,text="Make Payment of
    "+str(total_cost),bg='#ff8c00',fg='#57a1f8',font=('Microsoft
   YaHei UI Light',15,'bold','italic'),border=0,highlightbackground
   ="#ff8c00",command=lambda: select_payment_option(username)).
   place(x=35,y=120)
        print(total_cost)
```

10. **Payment Options:** The below method will ask user to select debit/credit or Cash for payment

```
def creditdebit(username):
    print("Display Credit Debit Option:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=440,y=70)
    heading = Label(frame, text='Almost Done!!!!!', fg='#57a1f8',bg
   ='white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    messagebox.showinfo("Payment Processing", "Payment is being
   processed...")
    # Insert code here to process the payment, e.g. call an API or
   perform necessary actions
    messagebox.showinfo("Payment Success", "Payment has been
   successfully processed!")
    Button(frame,width=25,pady=7,text="Print Invoice",bg='#ff8c00',
   fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'
   ),border=0,highlightbackground="#ff8c00",command=lambda:
   printinvoice(username)).place(x=40,y=107)
```

11. **Select payment screen:** The below method will display the payment screen

    - After selecting "Credit card/ Debit card" option user must enter his Debit/Credit card number. Here, if user will enter nothing and press "Proceed for Payment" then user will get popup like "invalid Input". Here, we handled the exception. So, if user enter incorrect card number, then system needs to stop user to make payment. So, we have been considered security concerns for payment as well. After entering correct card number user will get two popups like "Payment is being processed..." and after that "Payment has been successfully processed!". At the end user can also print invoice by clicking on "Print Invoice" button.

```
def select_payment_option(username):
    print("Display Payment Option:"+username)
    root.img = PhotoImage(file='login.png')
    Label(root,image=img,bg='white').place(x=50,y=50)
    frame = Frame(root,width=400,height=550,bg="white")
    frame.place(x=440,y=70)
    heading = Label(frame, text='Payment Options', fg='#57a1f8',bg='white',font=('Microsoft YaHei UI Light',23,'bold'))
    heading.place(x=100,y=5)
    Button(frame,width=25,pady=7,text="Credit/Debit Card",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00",command=lambda: creditdebit(username)).place(x=40,y=107)
    Button(frame,width=25,pady=7,text="Cash",bg='#ff8c00',fg='#57a1f8',font=('Microsoft YaHei UI Light',15,'bold','italic'),border=0,highlightbackground="#ff8c00").place(x=40,y=170)
```

12. **Print Invoice**: The below method will print the Invoice of the ticket after the successful payment by the user

```
def printinvoice(username):
    print("Display Print Invoice Option:"+username)
    messagebox.showinfo("Printing Your Invoice", "Printing is under process!!")
    # Insert code here to process the payment, e.g. call an API or perform necessary actions
    messagebox.showinfo("Printing Done", "Printing has been successfully completed!")
```

8.2 Exception Handling

1. - We handled several exceptions as well and kept several validations throughout the iGo system for security purpose. So, if user tries to press Login button without entering id which is “child or adult” in this case and password is “1234” then iGo will handle this exception by showing popup message which is “invalid username and password” or “invalid username” or “invalid password” based on the credentials enter by user.

```
def signin():
    username = user.get()
    password_input = password.get()
#     child
    if username == 'child' and password_input == '1234':
#         Select Language
        select_language_display(username)
    elif username == 'adult' and password_input == '1234':
        select_language_display(username)
    elif (username != 'child' or username != 'adult') and
   password_input != '1234':
        messagebox.showerror("Invalid","invalid username and password")
    elif password_input != '1234':
        messagebox.showerror("Invalid","invalid password")
    elif username != 'child':
        messagebox.showerror("Invalid","invalid username")
    elif username != 'adult':
        messagebox.showerror("Invalid","invalid username")
```

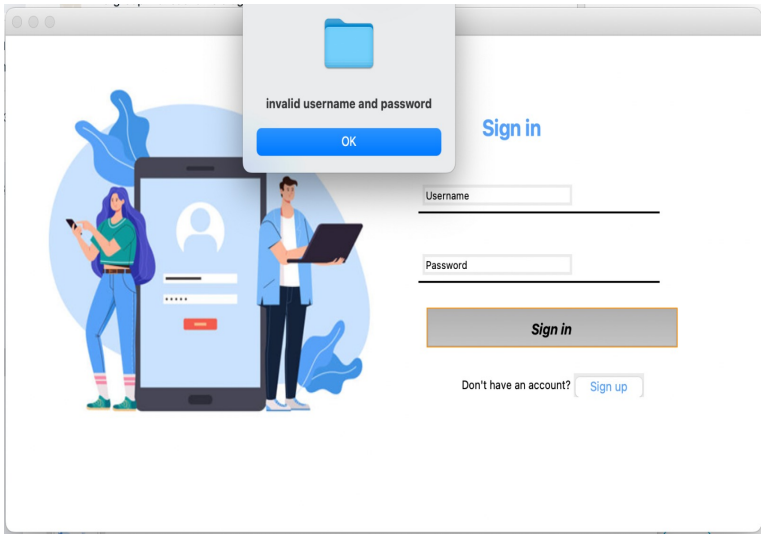

Figure 8.1: Validation for Username and Password

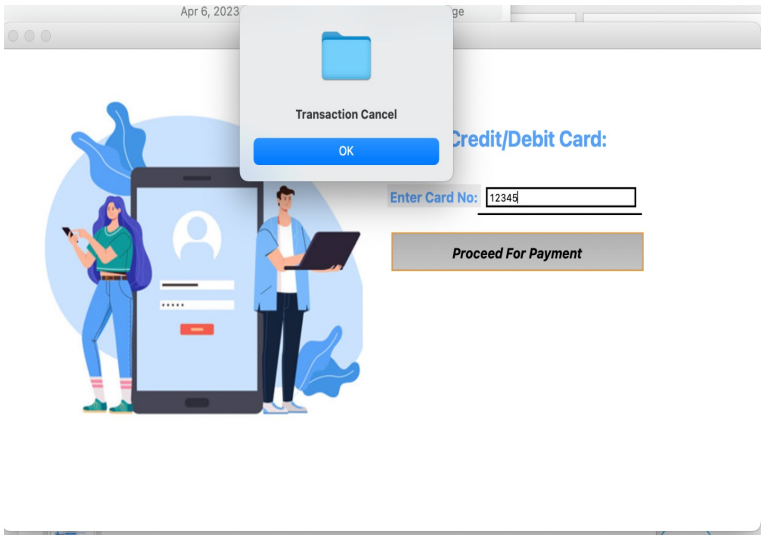

Figure 8.2: Debit/Credit Card Validation screen

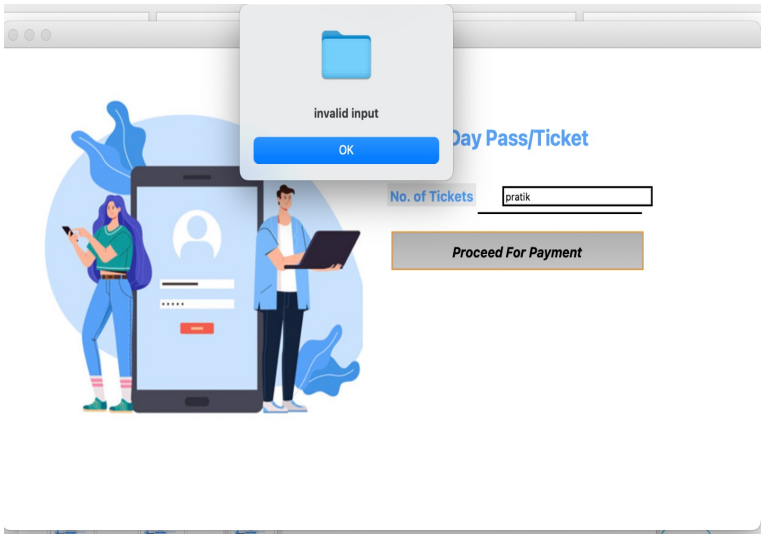

Figure 8.3: User Input Validation

Problem 9

9.1 Positive and Negative potential Uses of iGO:

Positive Potential uses of iGO:

1. **Convenience:** iGO can provide users with a convenient way to purchase tickets for events, shows, and concerts, without having to stand in long lines or visit a physical box office.

2. **Accessibility:** iGO can make it easier for users with disabilities or mobility issues to purchase tickets and attend events, as they can do so from the comfort of their own homes.

3. **Personalization:** iGO can provide users with personalized recommendations based on their interests, preferences, and past ticket purchases, making it easier for them to discover new events and shows they may be interested in.

4. **Discounts and promotions:** iGO can offer users exclusive discounts, promotions, and loyalty programs, making it more affordable for them to attend events and shows.

Negative Potential uses of iGO:

1. **Scalping:** iGO can be used by scalpers to purchase large quantities of tickets and resell them at inflated prices, making it harder for genuine fans to purchase tickets at face value.

2. **Fraud:** iGO can be used by scammers to sell fake tickets or steal personal and financial information from unsuspecting users, leading to financial loss and identity theft.

3. **Reselling:** iGO can be used by individuals to resell tickets for a profit, which may be illegal or violate the terms and conditions of the event or show.

4. **Monopoly:** iGO can be used by a single entity to monopolize the ticket sales for events and shows, leading to limited options and higher prices for consumers.

Screenshots of iGO that also corresponds to above mentioned Positive and Negative potential uses are attached in the below section.

9.2 Screenshots

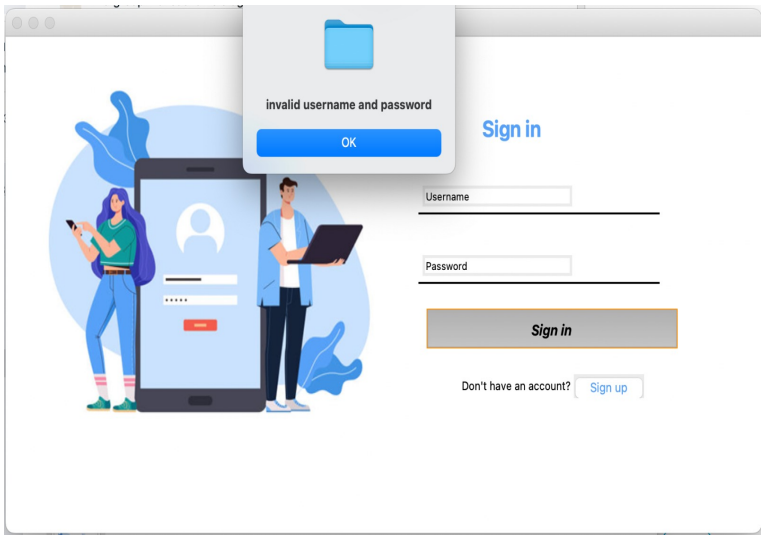

Figure 9.1: Main Page of the Application

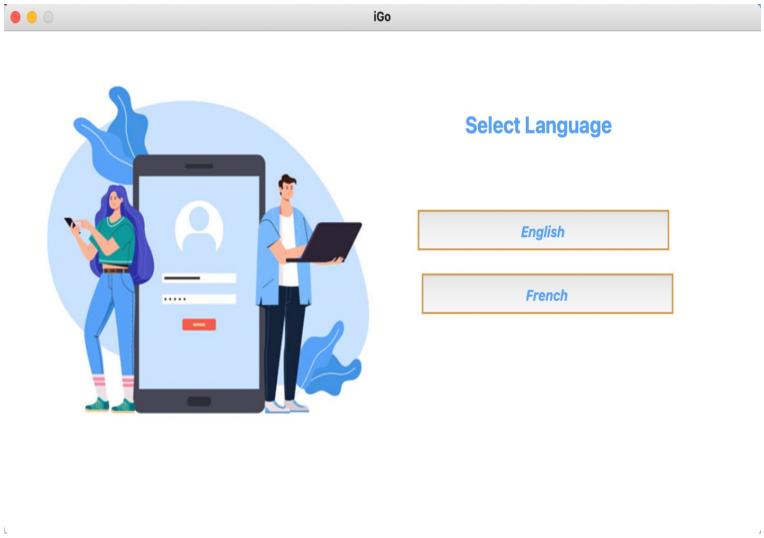

Figure 9.2: Language Selection screen

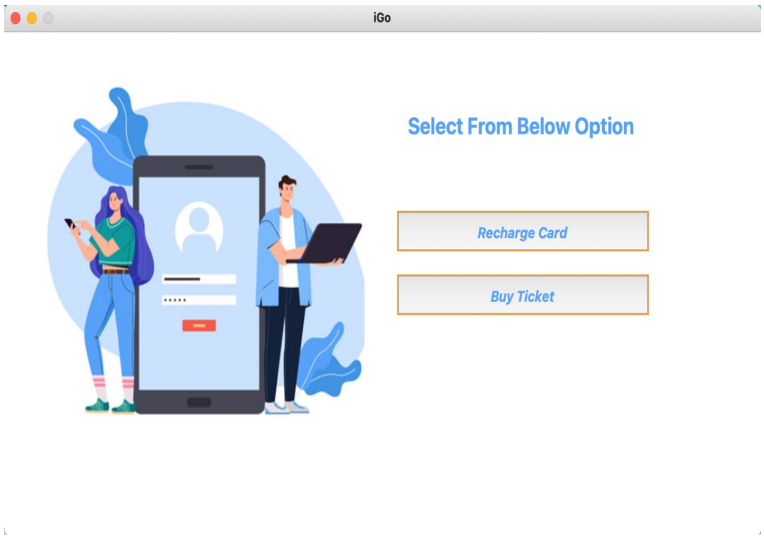

Figure 9.3: iGo - Menu screen

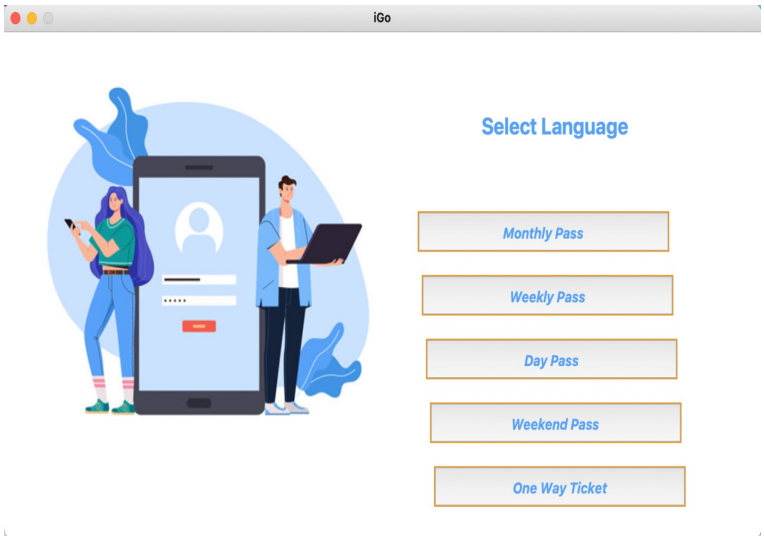

Figure 9.4: iGO - User Pass Selection screen

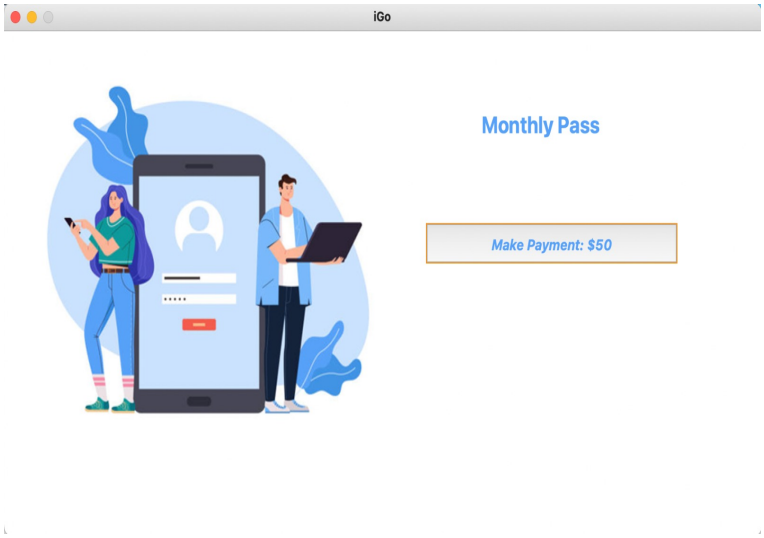

Figure 9.5: iGO - Monthly Pass Checkout screen

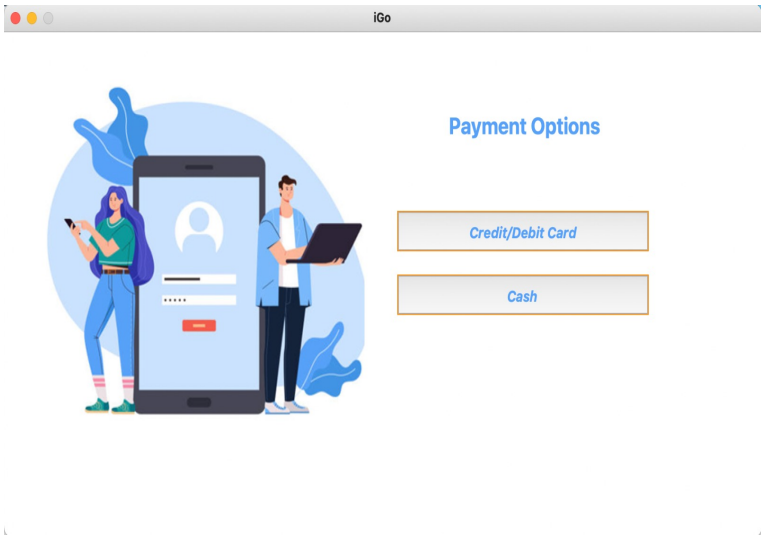

Figure 9.6: iGO - Payment screen

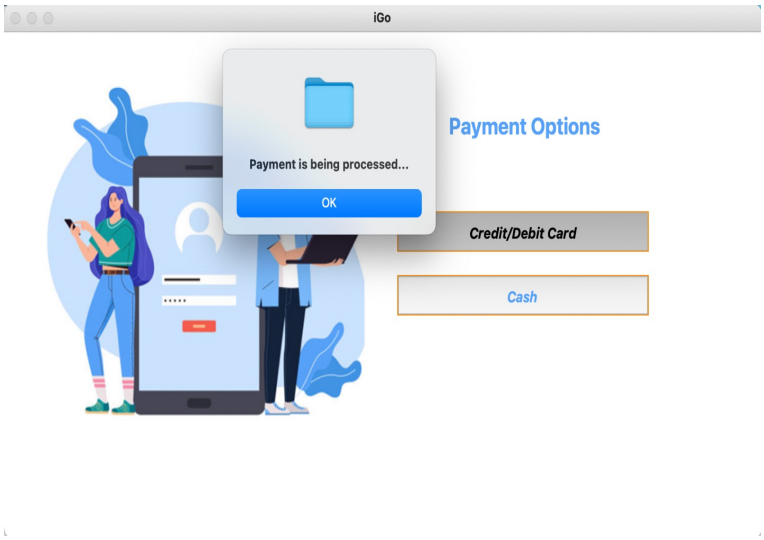

Figure 9.7: iGO - Payment Processing screen

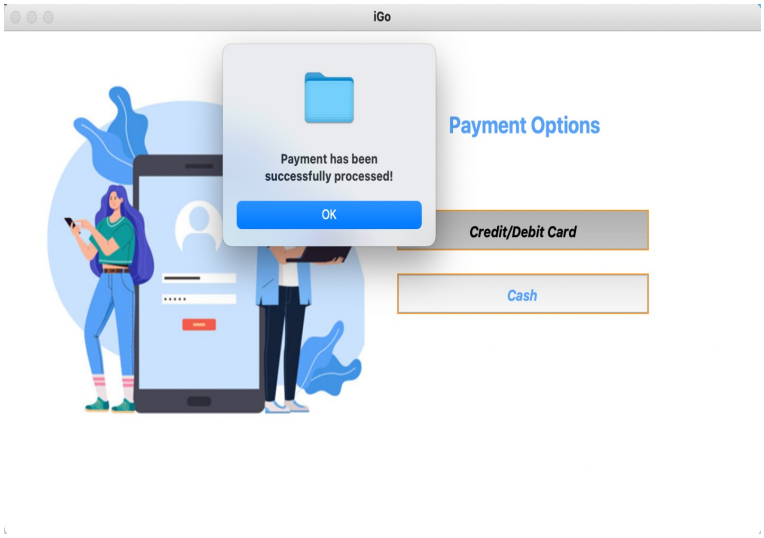

Figure 9.8: iGO - Payment Successful screen

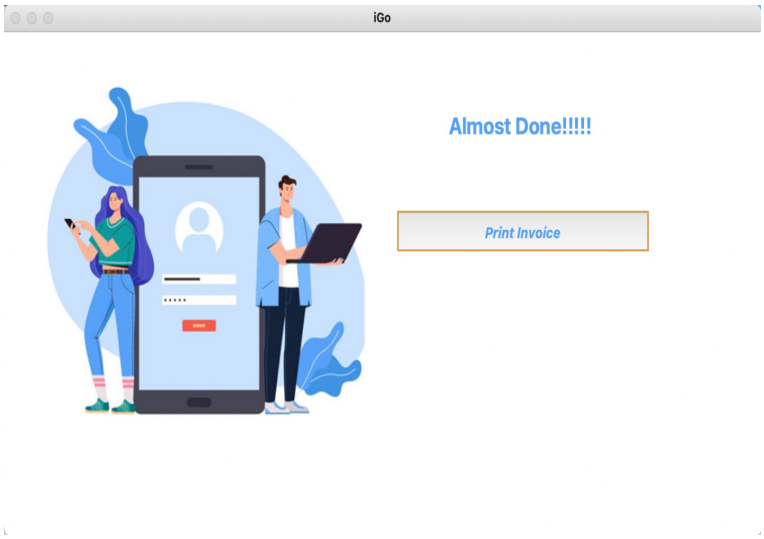

Figure 9.9: iGO - Print Invoice Screen

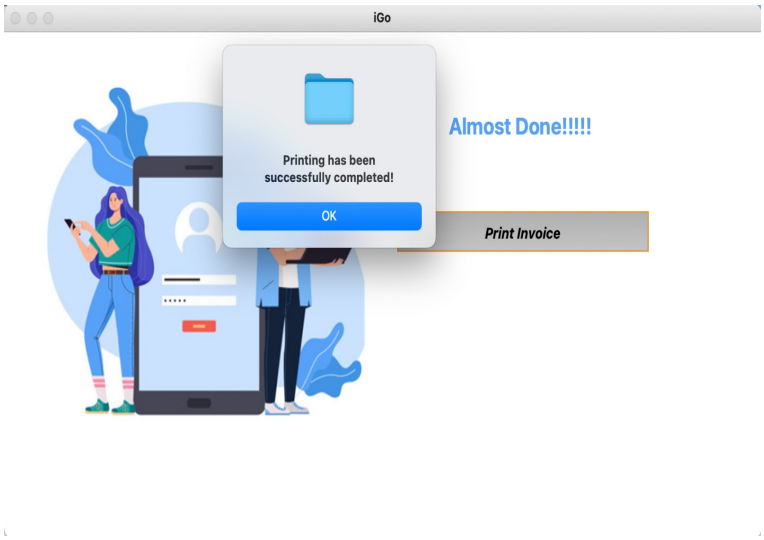

Figure 9.10: iGO - Print Successful screen

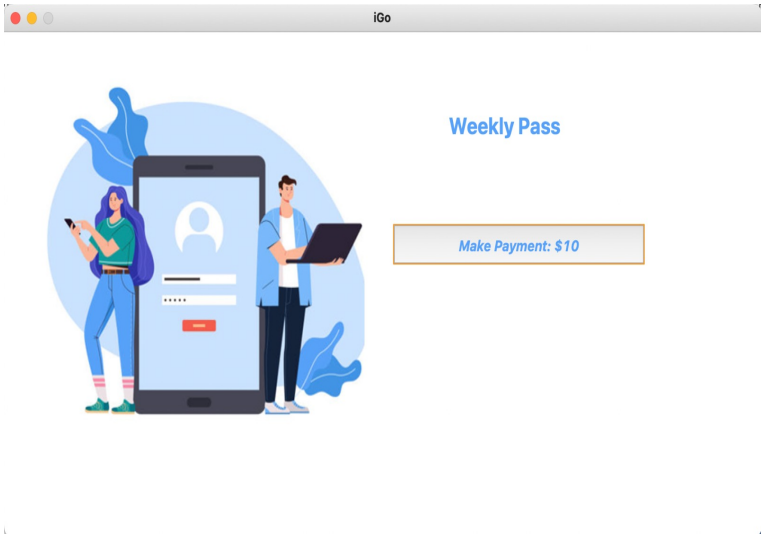

Figure 9.11: iGO - Weekly Pass Checkout screen

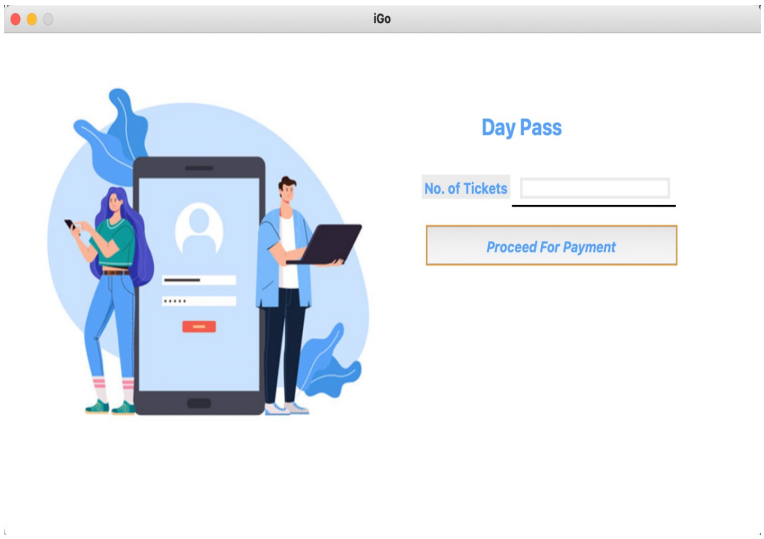

Figure 9.12: iGO - Day Pass Checkout screen

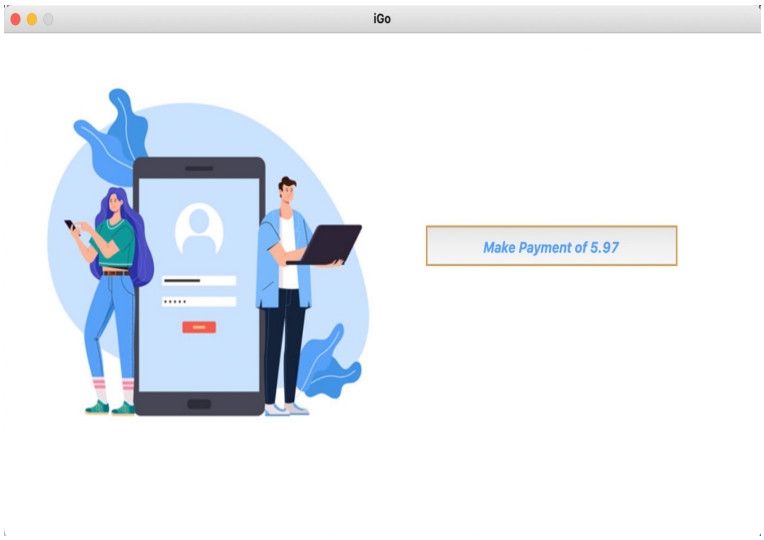

Figure 9.13: iGO - Day Pass Payment screen

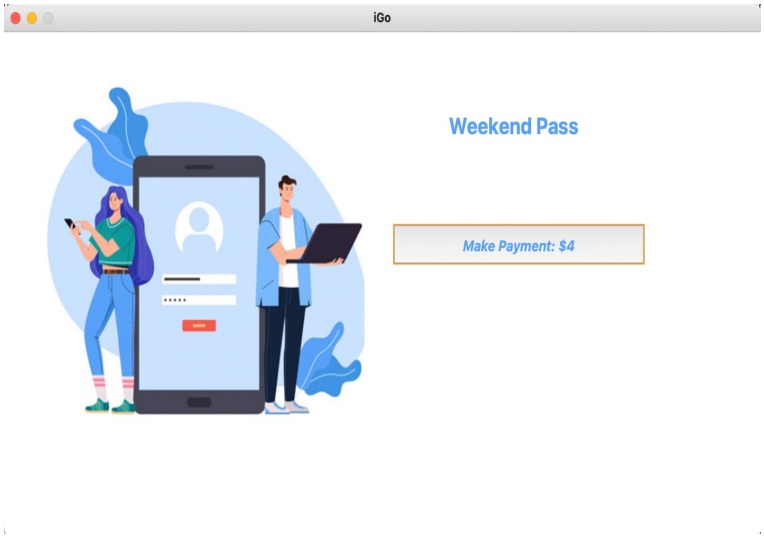

Figure 9.14: iGO - Weekend Pass Checkout screen

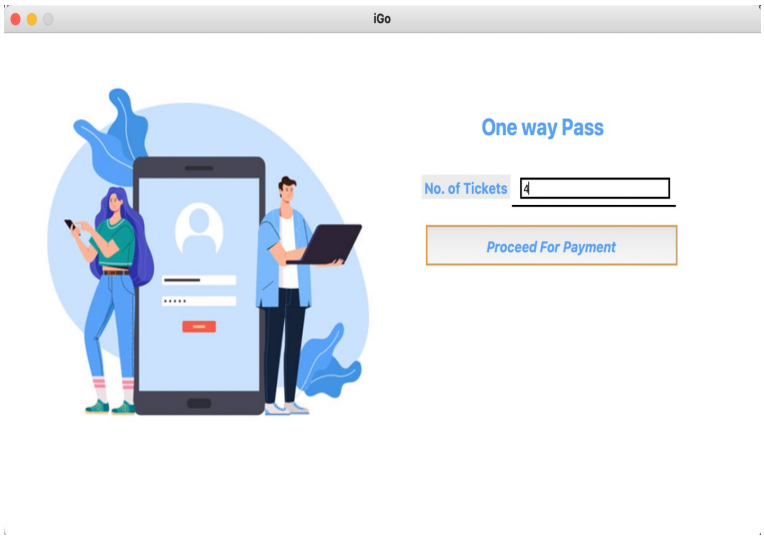

Figure 9.15: iGO - One-way Ticket Checkout screen

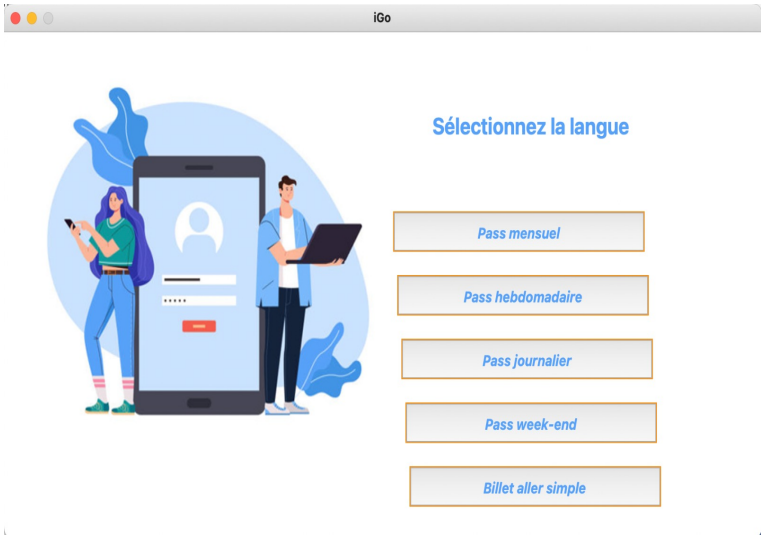

Figure 9.16: iGo - French Version

References

1. http://www.stm.info/en
2. PANKAJ KAMTHAN (2023) “Understanding Context”
3. PANKAJ KAMTHAN (2023) “Introduction to Interviews”
4. PANKAJ KAMTHAN (2023) “Introduction To Domain Modeling” - Section 14, 15 ,16.
5. PANKAJ KAMTHAN (2023) “Introduction To Use Case Modeling” - Section 11, 12.
6. PANKAJ KAMTHAN (2023) “Negative Use Case Modeling” - Section 8
7. https://www.lucidchart.com/pages/uml-activity-diagram
8. Carte OPUS. To obtain your photo OPUS card. 2023. http://www.carteopus.info/.
9. Code Formatting in LaTeX https://www.overleaf.com/learn/latex/Code_listing