

My name:	Abdulla Bin Amroo Hashim
My student number:	7571562
My email address:	hashim005@mymail.sim.edu.sg
Assignment number:	1

Design Requirements

- App Icon & Name

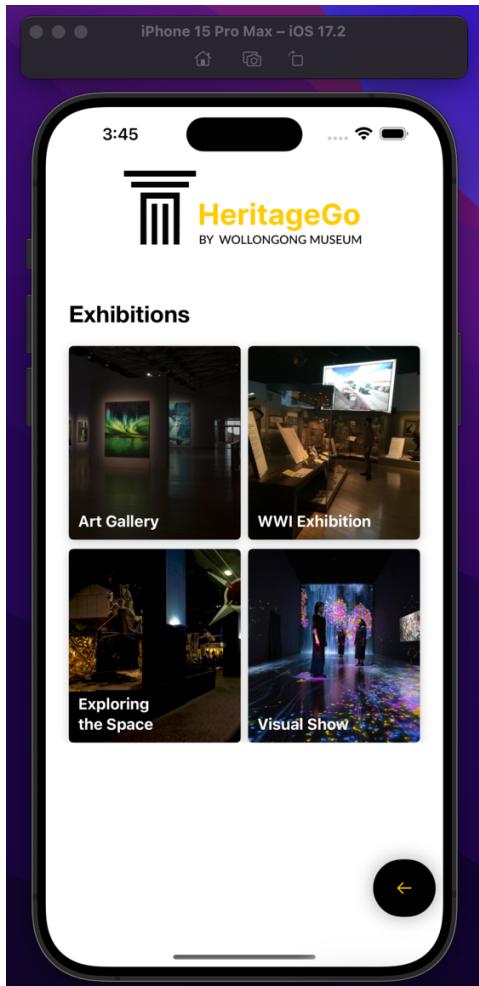


- Launch Screen and first page logo.

I created a custom swiftUI view (launchScreenView) and updated the SceneDelegate instead of utilizing the default launch screen to create a 2 second delay for the launch screen.



- Logo included in the First page.
- Each exhibitions presented by name and image.

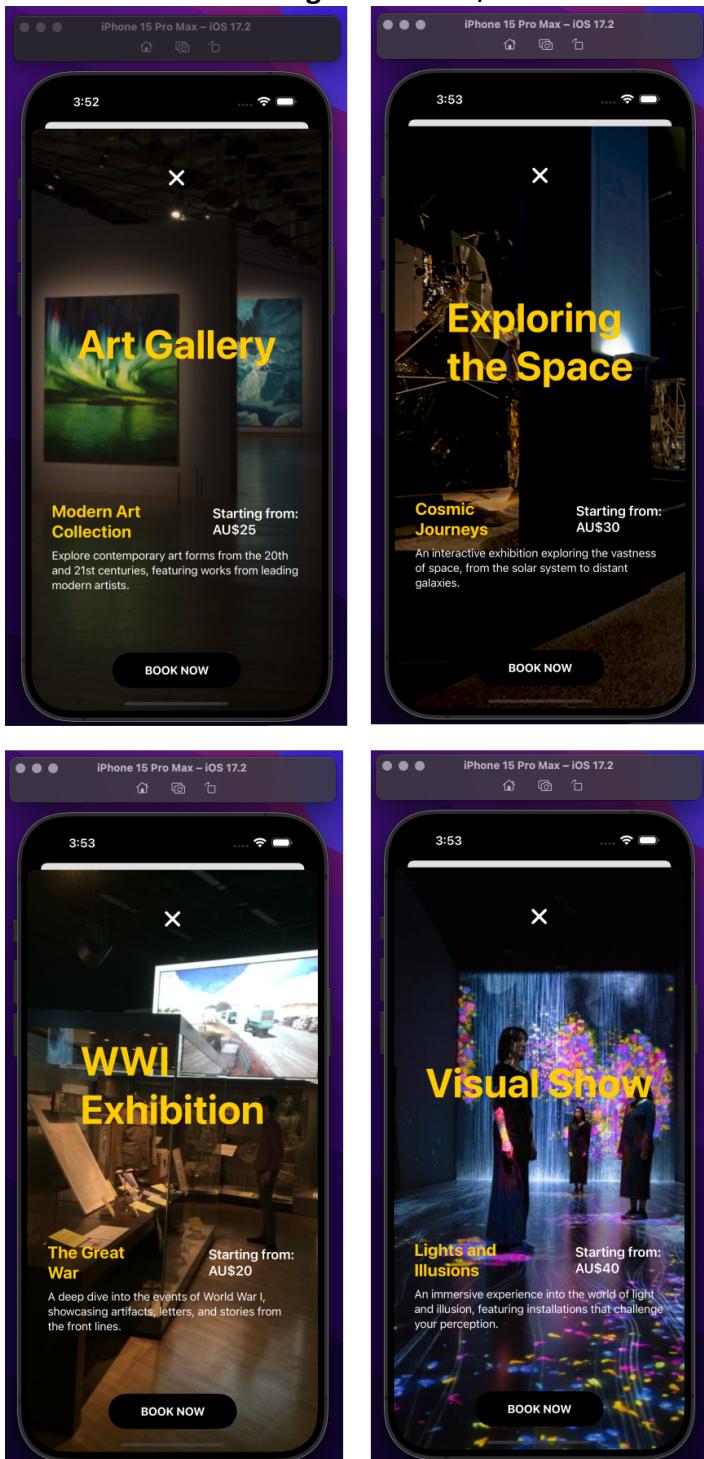


- Expandable floating tab bar for navigation, the bar is placed on right side for easier interactions via thumb in mind.

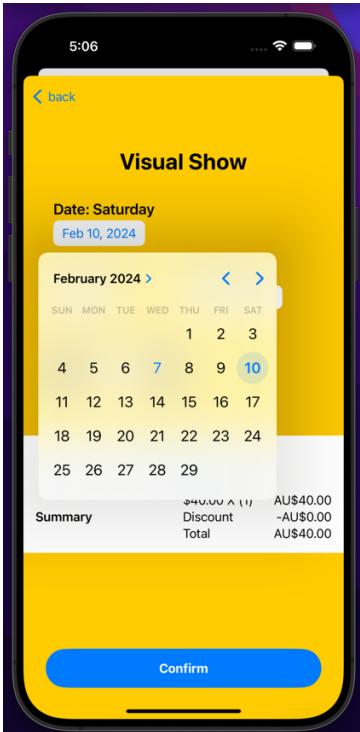


- By pressing cards in the exhibitions stack the user will be lead to Exhibitions Detail view which includes

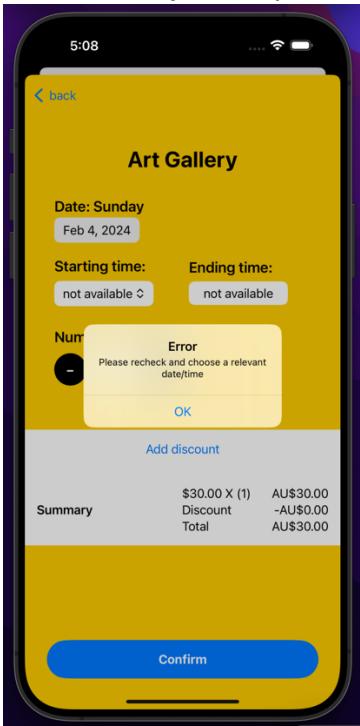
- I. Title
- II. Caption
- III. Description
- IV. Starting Price in AU\$



Date Picker (with Day shown in the Label above)



- If a day in the past is chosen the following alert will pop up.



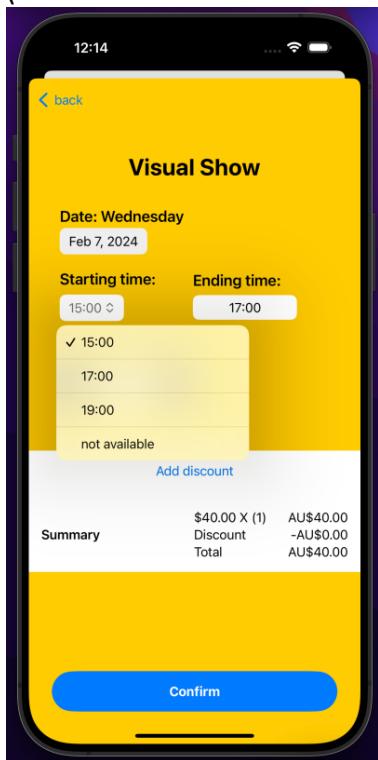
Starting Time Slot Availability:

```
390 // MARK: - Generate START Slots
391 private func generateTimeSlots(for event: String, on selectedDate: Date) -> [String] {
392     let calendar = Calendar.current
393     let now = Date()
394     let selectedDayStart = calendar.startOfDay(for: selectedDate)
395     let todayStart = calendar.startOfDay(for: now)
396
397     let isPastDate = selectedDayStart < todayStart
398     let isToday = calendar.isDateInToday(selectedDate)
399     let currentTime = calendar.component(.hour, from: now) * 60 + calendar.component(.minute, from: now) //
400
401     if isPastDate {
402
403         return []
404     } else {
405         var slots: [String] = []
406         if event == "Visual Show" {
407             let showSlots = ["15:00", "17:00", "19:00"]
408             if isToday {
409                 slots = showSlots.filter { slot in
410                     let slotHour = Int(slot.prefix(2)) ?? 0
411                     let slotMinute = Int(slot.suffix(2)) ?? 0
412                     let slotTimeInMinutes = slotHour * 60 + slotMinute
413                     return slotTimeInMinutes > currentTime
414                 }
415             } else {
416                 slots = showSlots
417             }
418         } else {
419             for hour in 9...19 {
420                 let slot1 = "\u{202e}(hour):00"
421                 let slot2 = "\u{202e}(hour):30"
422                 let slot1TimeInMinutes = hour * 60
423                 let slot2TimeInMinutes = hour * 60 + 30
424
425                 if !isToday || (isToday && slot1TimeInMinutes > currentTime) {
426                     slots.append(slot1)
427                 }
428                 if !isToday || (isToday && slot2TimeInMinutes > currentTime && hour < 20) {
429                     slots.append(slot2)
430                 }
431             }
432         }
433     }
434     return slots
435 }
```

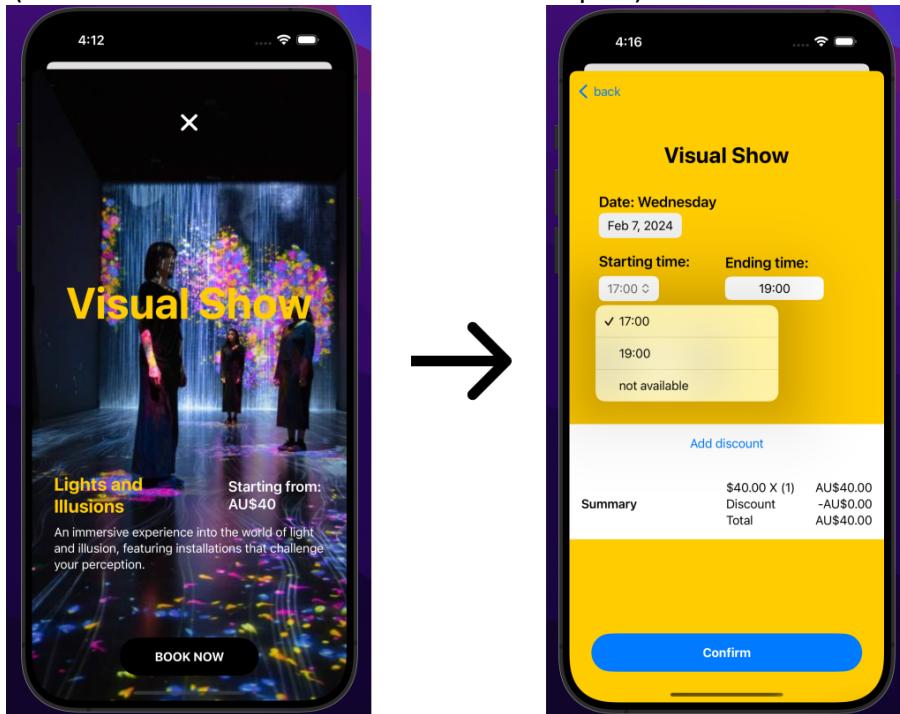
Note: All the exhibitions are functional and as per as requirements.

- **Starting Time slots for the Visual Show**

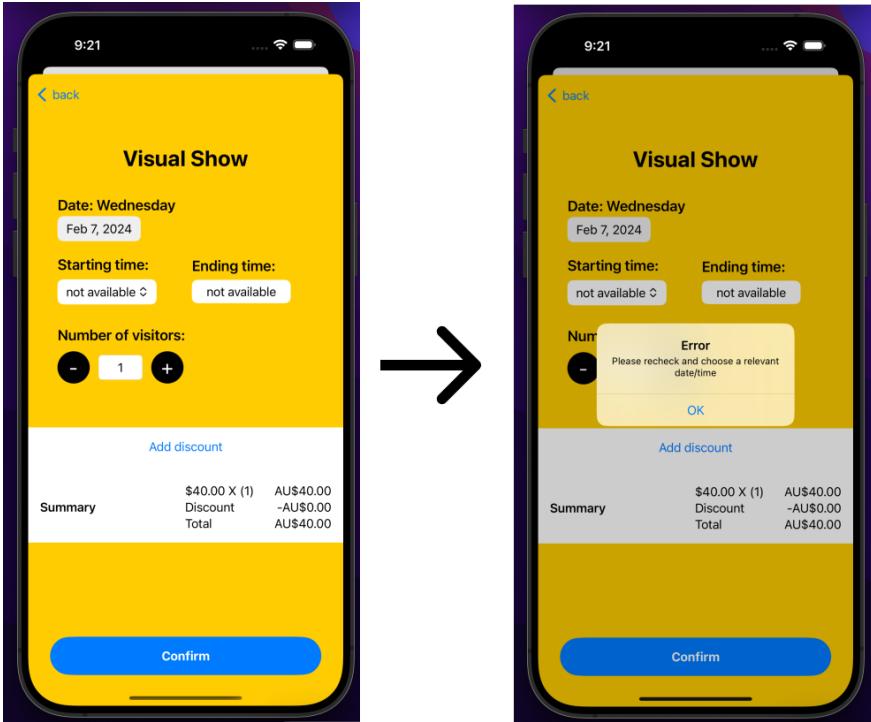
(All the time slots available depending on the time and day)



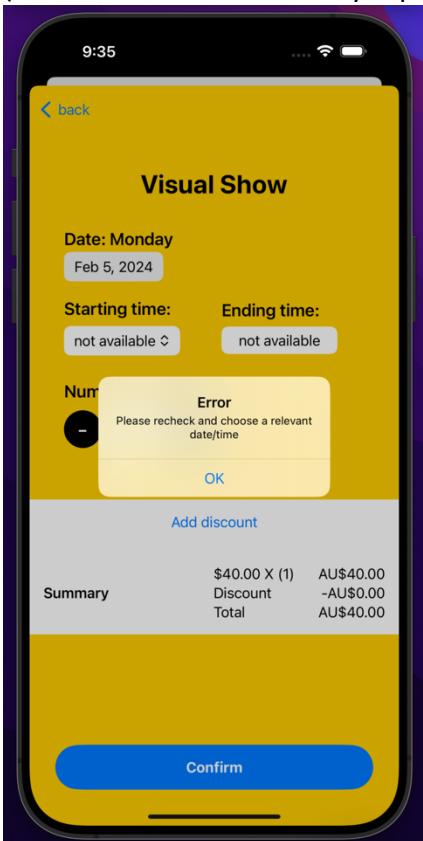
(Some time slots are not available if time is past)



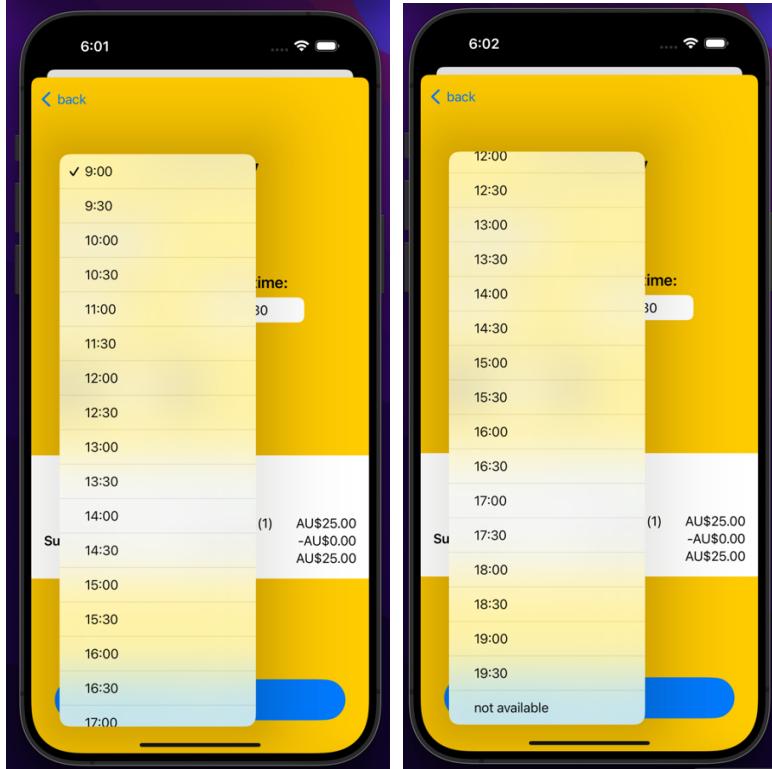
(no slots available if the time is past)



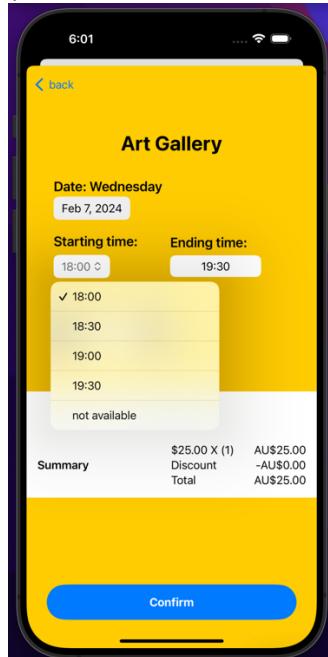
(no slots available if the day is past)



- **Starting Time slots for the Art Gallery, WWI Exhibition, Exploring the Space**
 (All the time slots available depending on the time and day every)



(Some time slots are not available if time is past)

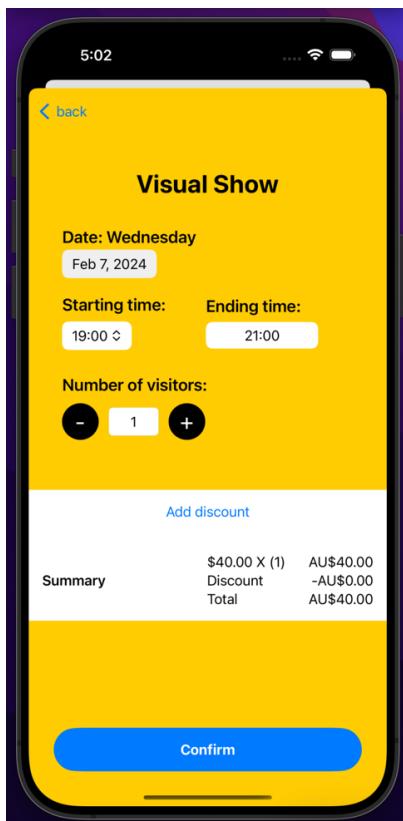


If no slots available if the day is past, no slots available if the time is past.
Same logic as Visual Show (Alert will pop up), consistent across all Exhibitions.

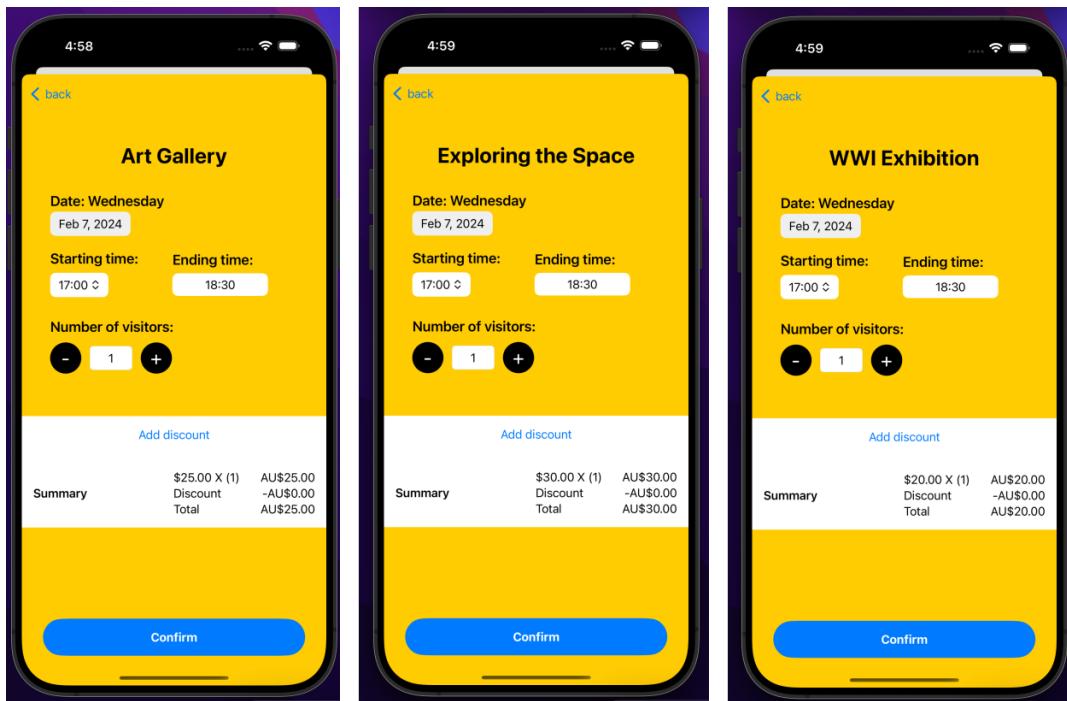
Ending Time Slot Calculation:

```
// MARK: - Update END TIME
func updateEndTime() {
    if startTime == "not available" {
        endTime = "not available"
    } else {
        let formatter = DateFormatter()
        formatter.dateFormat = "HH:mm"
        if let startDate = formatter.date(from: startTime) {
            var duration: TimeInterval = 0
            if event == "Visual Show" {
                duration = 2 * 60 * 60 // 2 hours in seconds
            } else {
                duration = 1.5 * 60 * 60 // 1.5 hours in seconds
            }
            let endDate = startDate.addingTimeInterval(duration)
            endTime = formatter.string(from: endDate)
        } else {
            endTime = "not available"
        }
    }
}
```

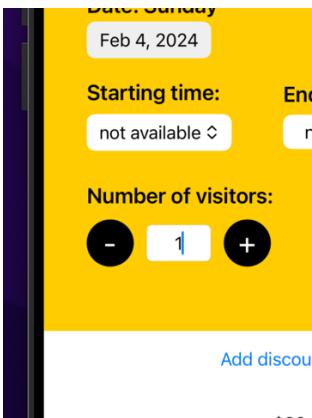
- Ending Time slots for the Visual Show (Every 2 hours)



- Ending Time slots for the Art Gallery, WWI Exhibition, Exploring the Space(Every 1.5 hours)



Stepper for Number of Visitors (with keyboard input on)



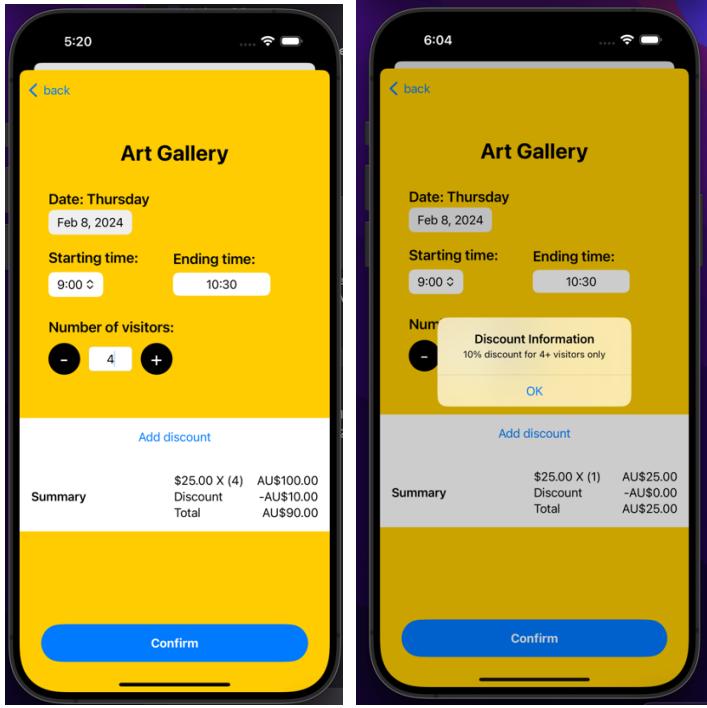
BookingView.swift stepper implementation.

```
155 // MARK: - Number of Visitors
156     VStack(alignment: .leading, spacing: 10) {
157         Text("Number of visitors:")
158             .fontWeight(.semibold)
159             .foregroundColor(.black)
160             .font(.system(size: 20))
161             .padding(.horizontal, 40)
162
163         HStack {
164             Button(action: {
165                 self.decrement()
166             }) {
167                 Text("-")
168                     .font(.title)
169                     .foregroundColor(.white)
170                     .frame(width: 44, height: 44)
171                     .background(Color.black)
172                     .clipShape(Circle())
173             }
174
175             TextField("Number", text: numberOfVisitorsString)
176                 .keyboardType(.numberPad)
177                 .frame(width: 60, height: 44)
178                 .textFieldStyle(RoundedBorderTextFieldStyle())
179                 .multilineTextAlignment(.center)
180                 .padding(.horizontal, 4)
181
182             Button(action: {
183                 self.increment()
184             }) {
185                 Text("+")
186                     .font(.title)
187                     .foregroundColor(.white)
188                     .frame(width: 44, height: 44)
189                     .background(Color.black)
190                     .clipShape(Circle())
191             }
192
193         }
194         .padding(.horizontal, 40)
195     }
196     .padding(.top, 20)
```

```
329     func increment() {
330         numberOfVisitors += 1
331     }
332
333
334
335     func decrement() {
336         if numberOfVisitors > 1 {
337             numberOfVisitors -= 1
338         }
339     }
340
```

Price and Discount as per required in \$AU.

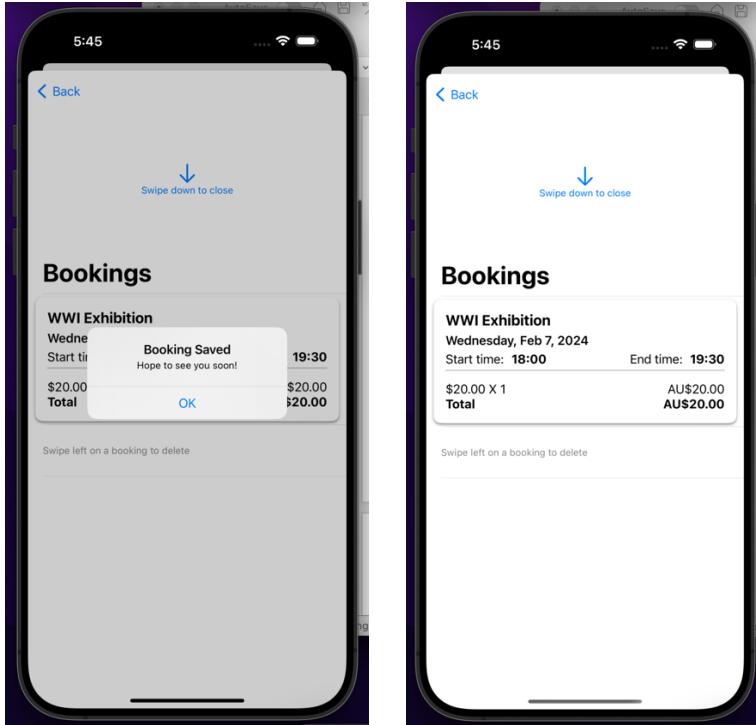
- If users are keen to get a discount and press the add discount button the following alert will pop up.



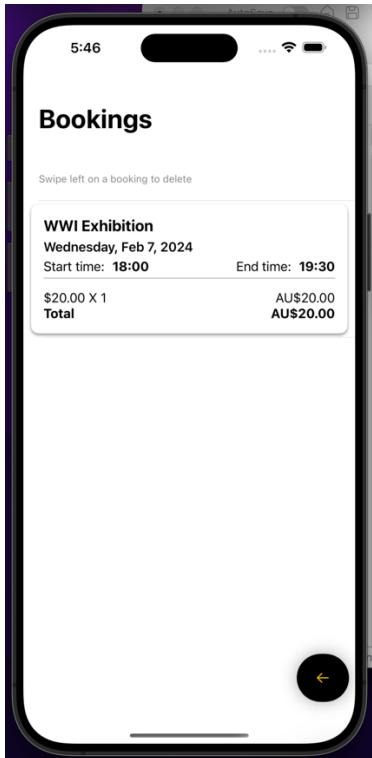
BookingView.swift pricing logic

```
365     // MARK: - Calculate Pricing
366     func calculatePricing() {
367         let prices: [String: (weekday: Double, weekend: Double)] = [
368             "Art Gallery": (weekday: 25, weekend: 30),
369             "WWI Exhibition": (weekday: 20, weekend: 25),
370             "Exploring the Space": (weekday: 30, weekend: 35),
371             "Visual Show": (weekday: 40, weekend: 40)
372         ]
373
374         let calendar = Calendar.current
375         let dayOfWeek = calendar.component(.weekday, from: selectedDate)
376         let isWeekend = dayOfWeek == 1 || dayOfWeek == 7
377
378         let eventPricing = prices[event] ?? (weekday: 0, weekend: 0)
379         let price = isWeekend ? eventPricing.weekend : eventPricing.weekday
380
381         pricePerVisitor = price
382
383
384         subtotal = price * Double(numberOfVisitors)
385         discount = numberOfRowsVisitors >= 4 ? subtotal * 0.1 : 0
386         total = subtotal - discount
387     }
388
389
```

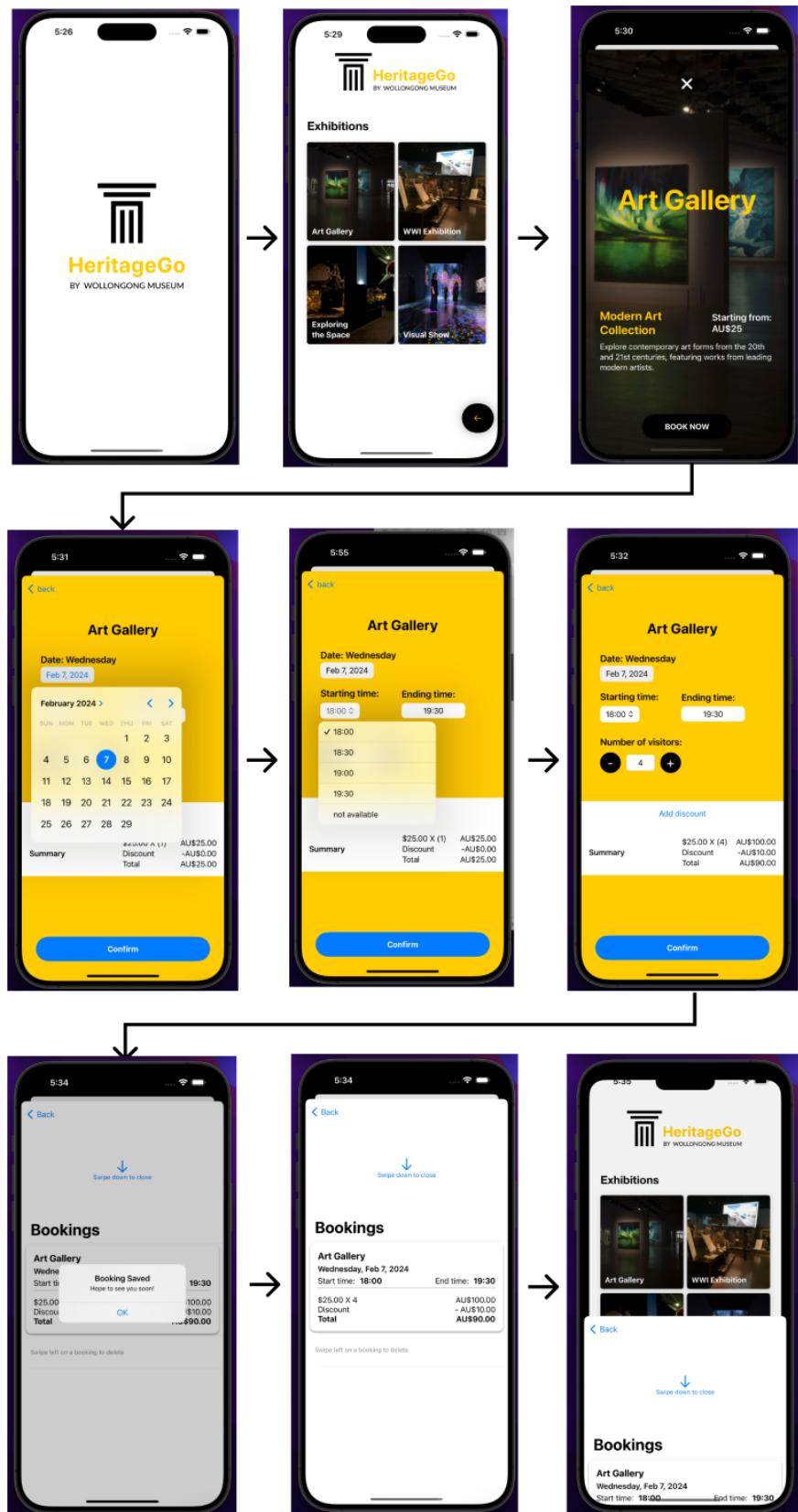
`savedBookingView2` after booking is confirmed.



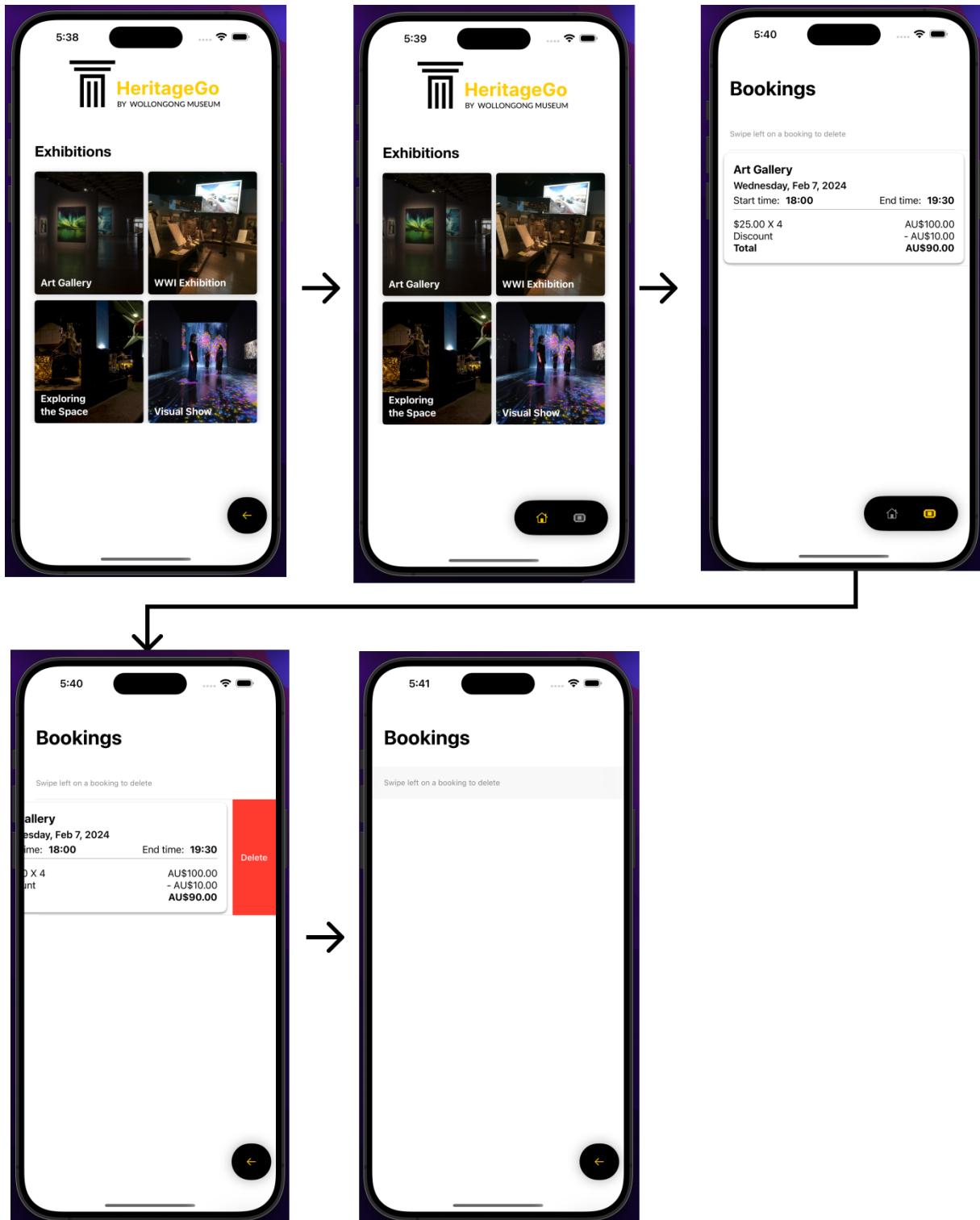
`savedBookingView1` navigated via floating tab bar.



Workflow



Navigation via floating tab bar



Examples Tested:

