Assignment 02 Presentation

CSC 220

Mos

Agenda

I do some talking:

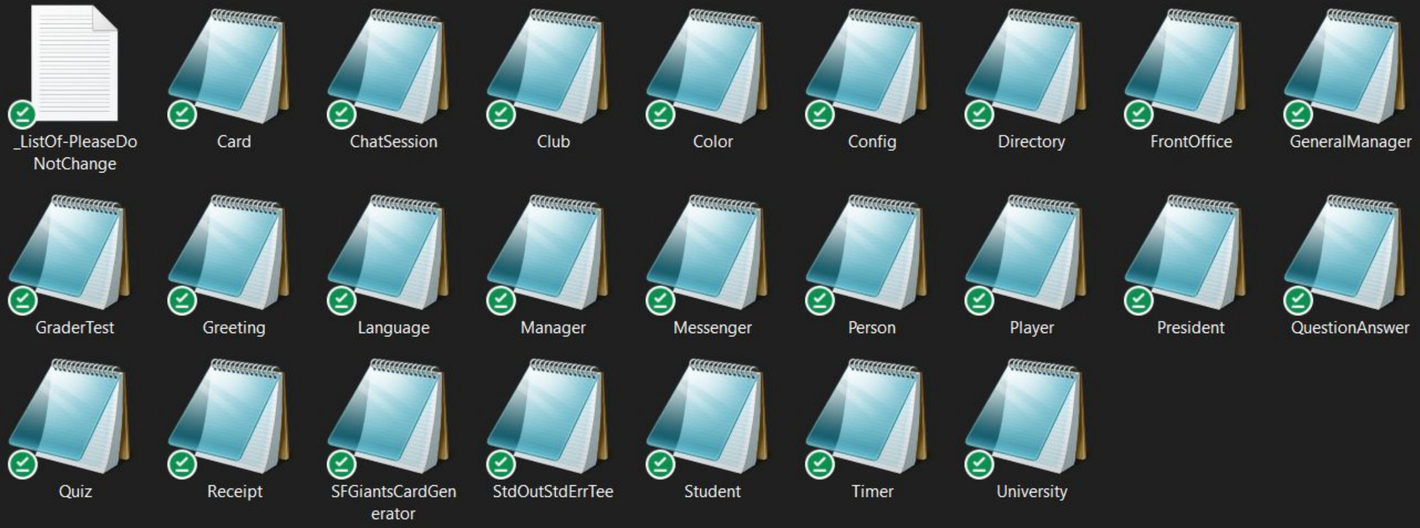
- Approach
- Logging and Testing
- Organization and Sustainability

Then,

- Demo
- Some Q&A, maybe

Give up and quit

- There's so many files in the folder,
  what on earth are we supposed to do with them?

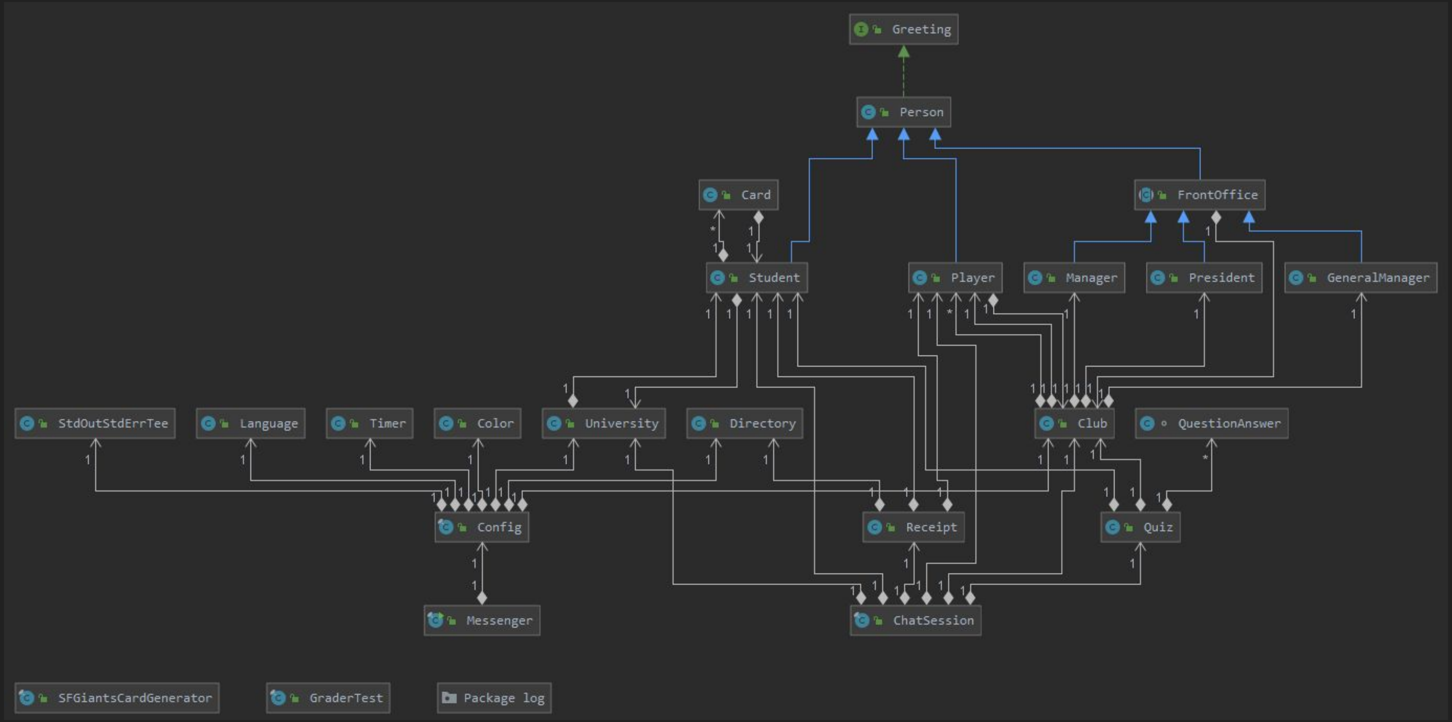


3 days
before the deadline

3 days
before the (first) deadline

Approach

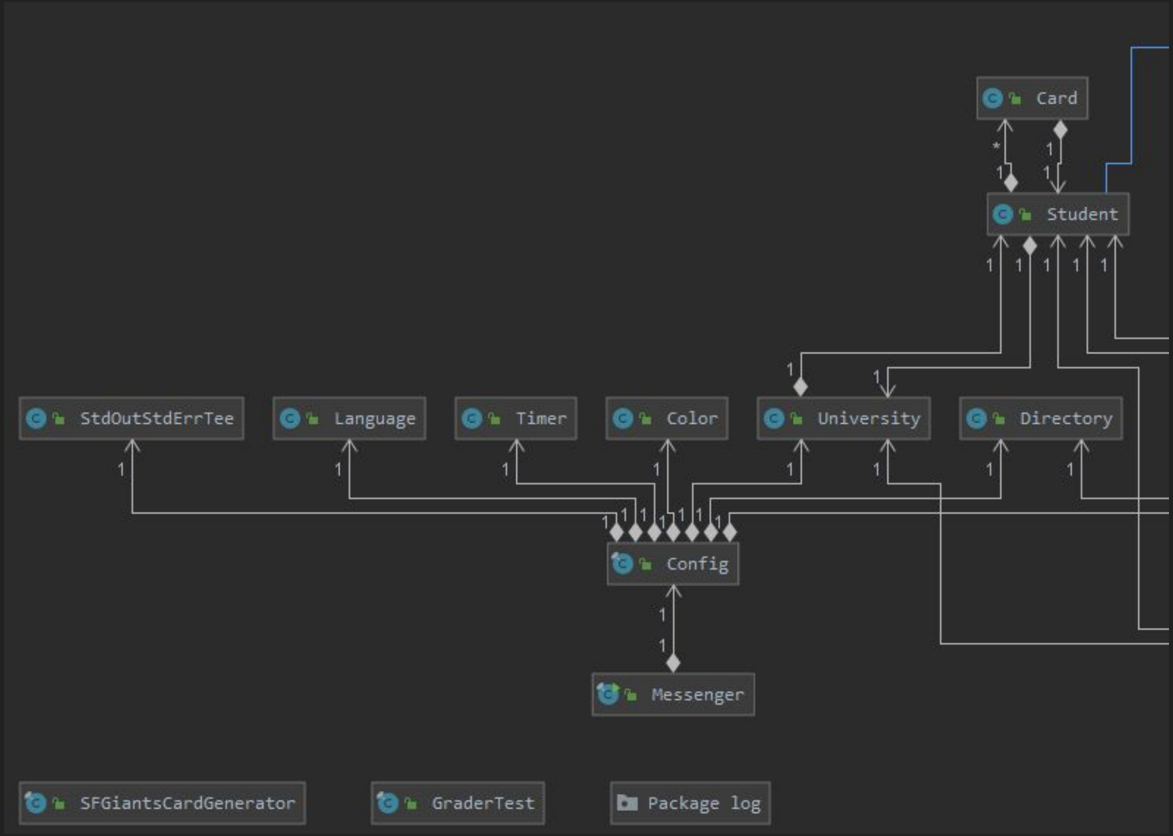
Remember this graph?



Started on the left-hand side

Logically, the program couldn't function if we haven't defined the configuration needed for it to run.

- This is where our main class is (Messenger).
- The main class depends on the Config class to handle the program's configuration.
- Each settings are handled by individual classes, such as language, time zone, log files, etc.



StdOutStdErrTee

Redirects the standard out and standard error stream to a custom stream that prints to console *and* writes to a file.

Creating file streams
Creates two file streams for standard output and standard error at the given path. Create streams at default path if creation at the specified path fails.

```java
private static FileOutputStream[] createFileStreams(String sOutPath,
String sErrPath) {
    try {
        return new FileOutputStream[]{
                new FileOutputStream(sOutPath),
                new FileOutputStream(sErrPath)
        };
    } catch (FileNotFoundException e) {
        System.err.printf(
                "Could not create file output streams at '%s' and '%s'.
Reason: %s\n Creating file output streams at default location instead...",
                sOutPath, sErrPath, e.getMessage()
        );
        return defaultFileStreams();
    }
}
```

StdOutStdErrTee

Redirects the standard out and standard error stream to a custom stream that prints to console *and* writes to a file.

Creating file streams
Creates two file streams for standard output and standard error at the given path. Create streams at default path if creation at the specified path fails.

```java
private static FileOutputStream[] defaultFileStreams() throws
UncheckedIOException {
   try {
       return new FileOutputStream[]{
               new FileOutputStream(Config.getDefaultStdOutFilePath()),
               new FileOutputStream(Config.getDefaultStdErrFilePath())
       };
   } catch (FileNotFoundException e) {
       throw new UncheckedIOException(e);
   }
}
```

StdOutStdErrTee

Redirects the standard out and standard error stream to a custom stream that prints to console *and* writes to a file.

Replacing default streams
Replaces the standard out and standard error streams with our custom stream that outputs to both the console and the file. Each time the streams are written to, the our overridden method is invoked.

```java
public void startLog() {
   StdOutStdErrTee sOutTee = new StdOutStdErrTee(
           System.out, (FileOutputStream) OUTPUT_STREAMS[0]);
   StdOutStdErrTee sErrTee = new StdOutStdErrTee(
           System.err, (FileOutputStream) OUTPUT_STREAMS[1]);
   PrintStream sOut = new PrintStream(sOutTee);
   PrintStream sErr = new PrintStream(sErrTee);

   System.setOut(sOut);
   System.setErr(sErr);
}

@Override
public void write(int b) throws IOException {
   for (OutputStream out : OUTPUT_STREAMS) {
       out.write(b);
       out.flush();
   }
}
```

Timer

Configures the time zone of the program.

Configuring the time zone
Looks up a time zone from the given name, then sets the time zone fields. Uses PST as fallback if a valid time zone cannot be found.

```
private final TimeZone timeZoneObject;
private final String timeZoneName;

// ...

public Timer(String timeZoneName) {
       this.timeZoneObject = findTimeZone(timeZoneName);
       this.timeZoneName = timeZoneObject.getDisplayName(isDst,
TimeZone.LONG);
   }
```

Timer

Configures the time zone of the program.

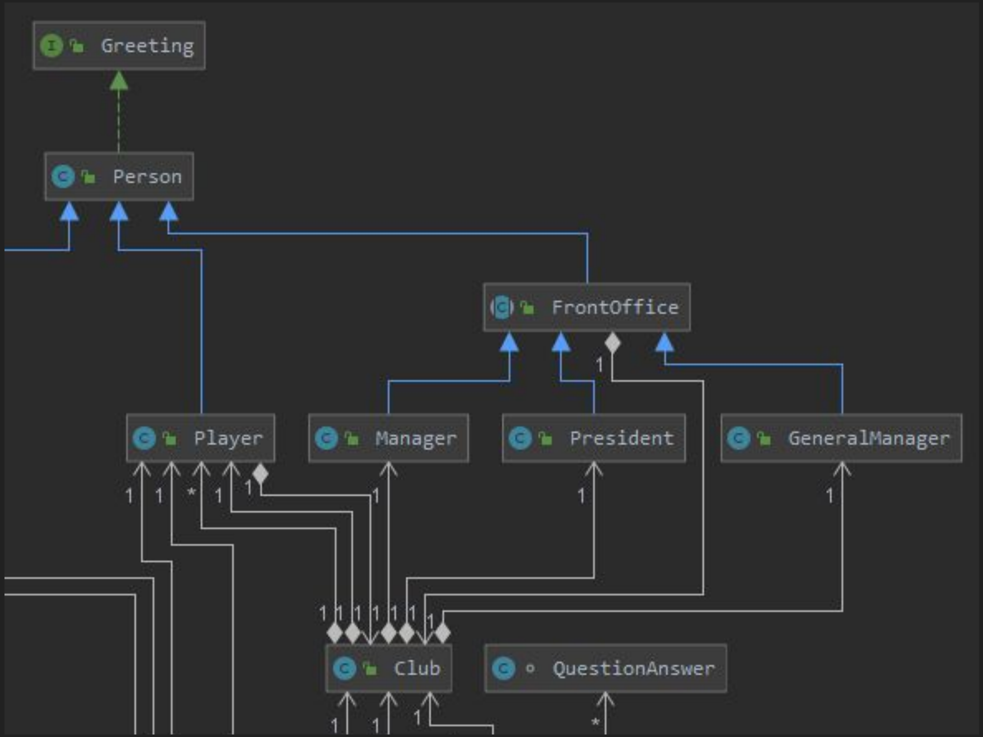
Looking up time zones
Searches a list of all time zones and its aliases. Returns a `TimeZone` object if one is found, otherwise returns a `TimeZone` object representing PST.

```java
private TimeZone findTimeZone(String q) {
   for (String supportedId : TimeZone.getAvailableIDs()) {
       TimeZone tz = TimeZone.getTimeZone(supportedId);

       // If it's the IANA ID.
       if (q.equalsIgnoreCase(supportedId)) {
           return tz;
       }
       // If it matches the daylight version of the name.
       if (q.equalsIgnoreCase(tz.getDisplayName(true, TimeZone.LONG))
               || q.equalsIgnoreCase(tz.getDisplayName(true,
                      TimeZone.SHORT))) {
           isDst = true;
           return tz;
       }
       // If it matches the non-daylight version of the name.
       if (q.equalsIgnoreCase(tz.getDisplayName(false, TimeZone.LONG))
               || q.equalsIgnoreCase(tz.getDisplayName(false,
                      TimeZone.SHORT))) {
           isDst = false;
           return tz;
       }
   }
   return DEFAULT_TZ;
}
```

Then, the top-right hand side

These appears to be added for “fake difficulty.” These classes didn’t do much apart from representing themselves.

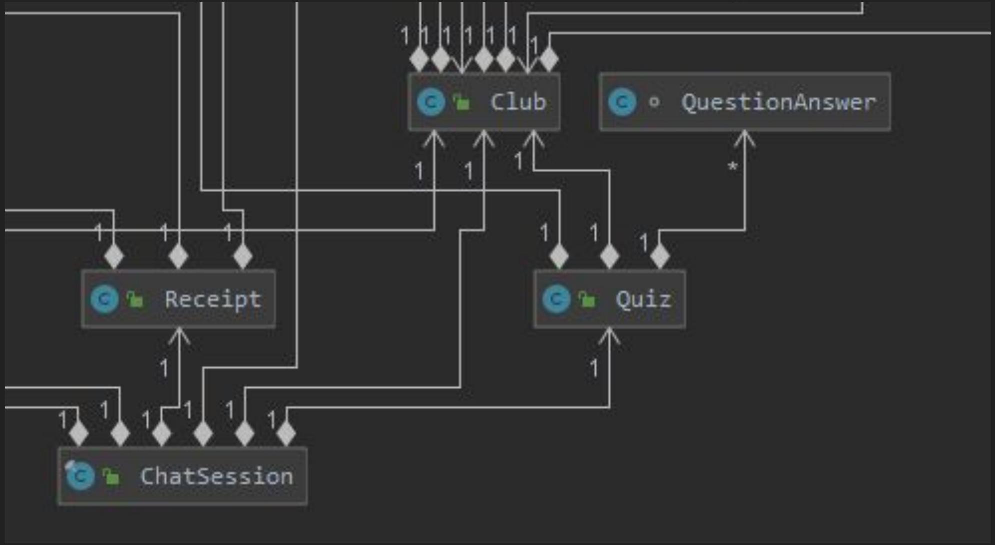
- Notice in the sample outputs, they only appear once at the start and never again.
- Only added the necessary fields and their getters and setters.



Finally,
bottom-right corner

The “brains” of the program.

- This is where the user actually interact with the program.
- Arguably, the more difficult ones to implement.
  - Tedious things to do like: input sanitation, validation, with internationalization on top, etc.



Logging and Testing

Logging

Not much choice here, since part of the requirement tells us how they want the logs to look like.

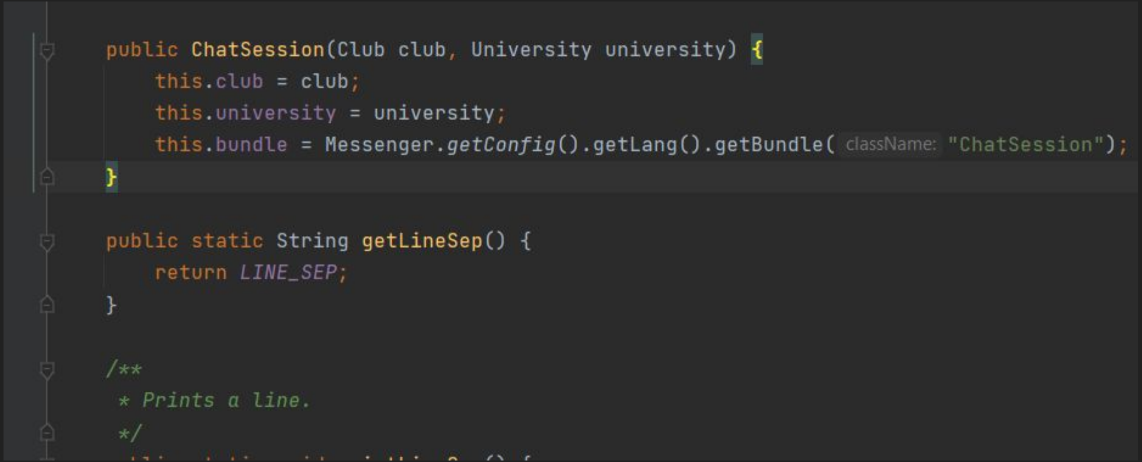
- Location can be configured in a file.
- Default location given in the Config class (not allowed to change).

```
Language:                 English
Time Zone:                Pacific Standard Time
Color:                    ANSI
Standard Output Log:      ./src/assignment02PartB/log/StandardOut.log
Standard Error Log:       ./src/assignment02PartB/log/StandardErr.log
Receipt Log:              ./src/assignment02PartB/log/Receipt-*-*.log
Default club:             San Francisco Giants
Default university:       San Francisco State University
------------------------------------------------------------------
2021/03/05 12:21:12 [0346 ms] PM PST - Chat session started.

SF Giants: Welcome to the SAN FRANCISCO GIANTS!
------------------------------------------------------------------
Club:                     San Francisco Giants
Short Name:               SF Giants
Established in:           1883
Colors:                   Orange, Black, Gold, Cream
Ballpark:                 Oracle Park
World Series Titles:      8
NL Pennants:              23
Division Titles:          8
Wild Card Berths:         3
Owners:                   San Francisco Baseball Associates LLC
President:                Farhan Zaidi
General Manager:          Scott Harris
Manager:                  Gabe Kapler
------------------------------------------------------------------
```

Testing

As a general rule, we test the code after finishing each method and classes to prevent more complicated problems later on.

- No time to write test classes for each and every classes and methods.
- Instead, we create a quick-and-dirty main method in the class to test the code we've written.



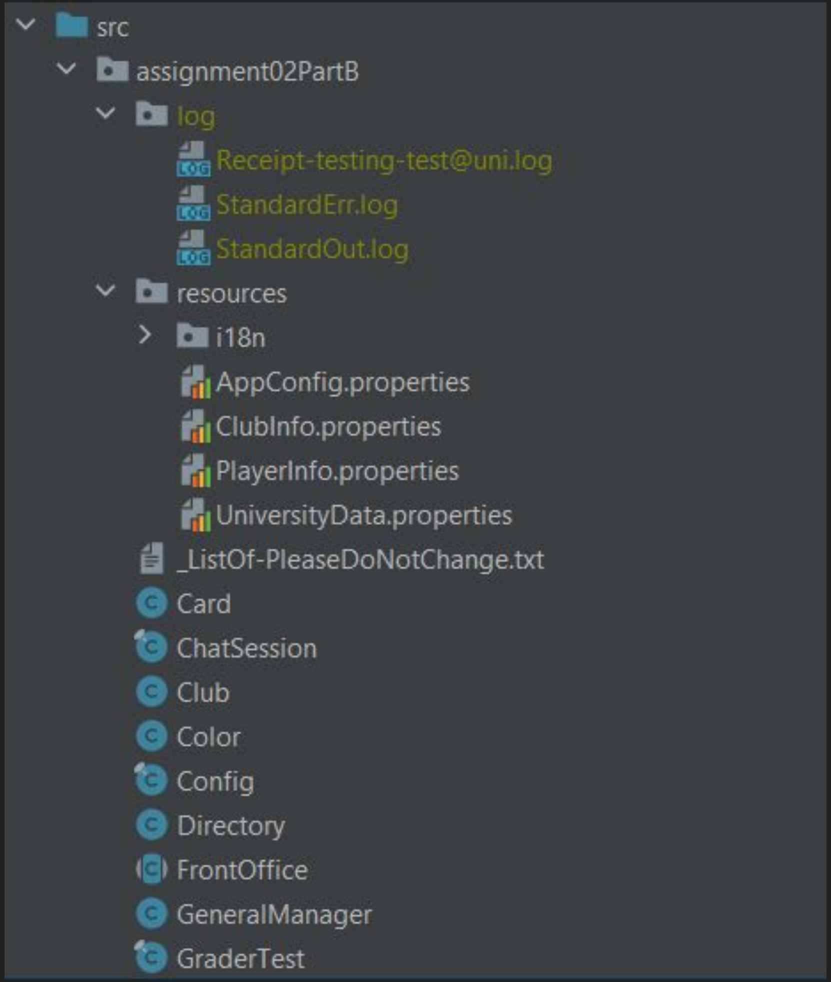
```
public ChatSession(Club club, University university) {
    this.club = club;
    this.university = university;
    this.bundle = Messenger.getConfig().getLang().getBundle( className: "ChatSession");
}

public static String getLineSep() {
    return LINE_SEP;
}

/**
 * Prints a line.
 */
```

Organization and Sustainability

Configurability

To make the program configurable, we store the configuration values outside of the source code.

- Uses Java's built-in [ResourceBundle]() module.
- Makes it much easier to configure later on.
- Avoid the risk of modifying the source code that may break the program.
    - It's also real messy towards the end 'cause I just rushed to finish it.



```
src
  assignment02PartB
    log
      Receipt-testing-test@uni.log
      StandardErr.log
      StandardOut.log
    resources
      i18n
      AppConfig.properties
      ClubInfo.properties
      PlayerInfo.properties
      UniversityData.properties
    _ListOf-PleaseDoNotChange.txt
    Card
    ChatSession
    Club
    Color
    Config
    Directory
    FrontOffice
    GeneralManager
    GraderTest
```

Configuring player information

Default file

```
firstName=Buster
lastName=Posey
club=San Francisco Giants
position=Catcher
number=28
batSide=Right
throwSide=Right
mlbDebut=2009
```

Updated file

```
firstName=Curt
lastName=Casali
club=San Francisco Giants
position=Catcher
number=2
batSide=Right
throwSide=Right
mlbDebut=2014
```

Internationalization and localization

Similar to how we configure the app, we also utilize the ResourceBundle module to add translation files.

- None of the strings in the interface are hardcoded into the classes.
- The ResourceBundle module loads the appropriate file for us, according to the language set at the start.
- ResourceBundle also handles fallbacks automatically.

```
------------------------------------------------------------------
------------------- SF GIANTS THANK YOU --------------------------
------------------------------------------------------------------

Language: en
Time Zone: pst
------------------------------------------------------------------
Language:                 English
Time Zone:                Pacific Standard Time
Color:                    ANSI
Standard Output Log:      ./src/assignment02PartB/log/StandardOut.log
Standard Error Log:       ./src/assignment02PartB/log/StandardErr.log
Receipt Log:              ./src/assignment02PartB/log/Receipt-*-*.log
Default club:             San Francisco Giants
Default university:       San Francisco State University
------------------------------------------------------------------
2021/03/31 09:45:57 [0744 ms] PM PST - Chat session started.
```

```
------------------------------------------------------------------
------------------- SF GIANTS THANK YOU --------------------------
------------------------------------------------------------------

Language: th
โซนเวลา: ict
------------------------------------------------------------------
ภาษา:                     Thai
โซนเวลา:                   Indochina Time
สี:                         ANSI
ล็อก Standard Output:     ./src/assignment02PartB/log/StandardOut.log
ล็อก Standard Error:      ./src/assignment02PartB/log/StandardErr.log
ล็อกใบเสร็จ:                ./src/assignment02PartB/log/Receipt-*-*.log
ค่าเริ่มต้นสำหรับชมรม:            ซานฟรานซิสโกใจแอนท์
ค่าเริ่มต้นสำหรับมหาวิทยาลัย:        มหาวิทยาลัยซานฟรานซิสโกสเตท
------------------------------------------------------------------
2021/04/01 11:48:39 [0893 ms] AM ICT - Chat session started.
```

Localizing the start menu

resources/i18n/Config.properties

```
# Default strings for displaying preferences
at the start (en_US)
language.label=Language
timeZone.label=Time Zone
color.label=Color
stdOutLogPath.label=Standard Output Log
stdErrLogPath.label=Standard Error Log
receiptLogPath.label=Receipt Log
defaultClub.label=Default club
defaultUniversity.label=Default university
defaultClub.value=San Francisco Giants
defaultUniversity.value=San Francisco State
University
```

resources/i18n/Config_**th**.properties

```
# Default strings for displaying preferences
at the start
language.label=ภาษา
timeZone.label=โซนเวลา
color.label=สี
stdOutLogPath.label=ล็อก Standard Output
stdErrLogPath.label=ล็อก Standard Error
receiptLogPath.label=ล็อกใบเสร็จ
defaultClub.label=ค่าเริ่มต้นสำหรับชมรม
defaultUniversity.label=ค่าเริ่มต้นสำหรับมหาวิทยาลัย
defaultClub.value=ซานฟรานซิสโกใจแอนท์
defaultUniversity.value=มหาวิทยาลัยซานฟรานซิสโกส
เตท
```

Localizing the start menu

resources/i18n/Config.properties

```
# Default strings for displaying preferences
at the start (en_US)
language.label=Language
timeZone.label=Time Zone
color.label=Color
stdOutLogPath.label=Standard Output Log
stdErrLogPath.label=Standard Error Log
receiptLogPath.label=Receipt Log
defaultClub.label=Default club
defaultUniversity.label=Default university
defaultClub.value=San Francisco Giants
defaultUniversity.value=San Francisco State
University
```

resources/i18n/Config_**alien**.properties

```
# Default strings for displaying preferences
at the start
language.label=#AL~~~ ~~~ AL#
timeZone.label=#AL~~~ ~~~ AL#
color.label=#AL~~~ ~~~ AL#
stdOutLogPath.label=#AL~~~ ~~~ AL#
stdErrLogPath.label=#AL~~~ ~~~ AL#
receiptLogPath.label=#AL~~~ ~~~ AL#
defaultClub.label=#AL~~~ ~~~ AL#
defaultUniversity.label=#AL~~~ ~~~ AL#
defaultClub.value=#AL~~~ ~~~ AL#
defaultUniversity.value=#AL~~~ ~~~ AL#
```

Demo

And that’s it!

Thanks for attending the presentation.
Please let me know if you have any questions.

Slides and code:
github.com/mosguinz-csc220-02/CSC220Asmt02