A screenshot of a computer program

Description automatically generated

A group of white rectangular boxes with text

Description automatically generated with medium confidence

@startuml

' Boundary Classes

class project\_help\_boundary <<Boundary>> {

+commands: List<String>

+project\_help()

}

class receive\_email\_boundary <<Boundary>> {

+commands: List<String>

+receive\_email()

}

class close\_browser\_boundary <<Boundary>> {

+commands: List<String>

+close\_browser()

}

class login\_boundary <<Boundary>> {

+commands: List<String>

+login()

}

class navigate\_to\_website\_boundary <<Boundary>> {

+commands: List<String>

+navigate\_to\_website()

}

class check\_availability\_boundary <<Boundary>> {

+commands: List<String>

+check\_availability()

}

class start\_monitoring\_availability\_boundary <<Boundary>> {

+commands: List<String>

+start\_monitoring\_availability()

}

class stop\_monitoring\_availability\_boundary <<Boundary>> {

+commands: List<String>

+stop\_monitoring\_availability()

}

class get\_price\_boundary <<Boundary>> {

+commands: List<String>

+get\_price()

}

class start\_monitoring\_price\_boundary <<Boundary>> {

+commands: List<String>

+start\_monitoring\_price()

}

class stop\_monitoring\_price\_boundary <<Boundary>> {

+commands: List<String>

+stop\_monitoring\_price()

}

' Control Classes

class project\_help\_control <<Control>> {

+execute()

}

class receive\_email\_control <<Control>> {

+execute()

}

class close\_browser\_control <<Control>> {

+execute()

}

class login\_control <<Control>> {

+execute()

}

class navigate\_to\_website\_control <<Control>> {

+execute()

}

class check\_availability\_control <<Control>> {

+execute()

}

class start\_monitoring\_availability\_control <<Control>> {

+execute()

}

class stop\_monitoring\_availability\_control <<Control>> {

+execute()

}

class get\_price\_control <<Control>> {

+execute()

}

class start\_monitoring\_price\_control <<Control>> {

+execute()

}

class stop\_monitoring\_price\_control <<Control>> {

+execute()

}

' Entity Classes

class AvailabilityEntity <<Entity>> {

+availability\_data: List<Data>

+last\_checked: DateTime

+check\_availability()

+export\_data()

}

class BrowserEntity <<Entity>> {

+cookies: List<Cookie>

+session\_data: Map<String, String>

+launch\_browser()

+navigate\_to\_website()

+close\_browser()

+login()

}

class DataExportEntity <<Entity>> {

+file\_paths: List<String>

+export\_to\_excel()

+export\_to\_html()

}

class EmailEntity <<Entity>> {

+email\_queue: List<Email>

+send\_email\_with\_attachments()

}

class PriceEntity <<Entity>> {

+price\_data: Map<String, Double>

+last\_updated: DateTime

+get\_price()

+export\_data()

}

' Relationships

project\_help\_boundary --> project\_help\_control

receive\_email\_boundary --> receive\_email\_control

close\_browser\_boundary --> close\_browser\_control

login\_boundary --> login\_control

navigate\_to\_website\_boundary --> navigate\_to\_website\_control

check\_availability\_boundary --> check\_availability\_control

start\_monitoring\_availability\_boundary --> start\_monitoring\_availability\_control

stop\_monitoring\_availability\_boundary --> stop\_monitoring\_availability\_control

get\_price\_boundary --> get\_price\_control

start\_monitoring\_price\_boundary --> start\_monitoring\_price\_control

stop\_monitoring\_price\_boundary --> stop\_monitoring\_price\_control

receive\_email\_control --> EmailEntity

close\_browser\_control --> BrowserEntity

login\_control --> BrowserEntity

navigate\_to\_website\_control --> BrowserEntity

check\_availability\_control --> AvailabilityEntity

start\_monitoring\_availability\_control --> AvailabilityEntity

stop\_monitoring\_availability\_control --> AvailabilityEntity

get\_price\_control --> PriceEntity

start\_monitoring\_price\_control --> PriceEntity

stop\_monitoring\_price\_control --> PriceEntity

check\_availability\_control --> DataExportEntity

get\_price\_control --> DataExportEntity

@enduml