

# Team 27 - Hagspár

Jón Guðjónsson

Sigurjón Ólafsson

Þorsteinn Sigurðsson

# Project inception and planning

# Hagspár

- Last year's project was a web app that allowed users to automatically create economic forecasts using input data from Statistics Iceland.
- App extends the capabilities of the web app from last year natively to a mobile platform
- Adds value by allowing desktop users to create and review their forecasts on the go (without having to use unwieldy web app on their phones' browser).

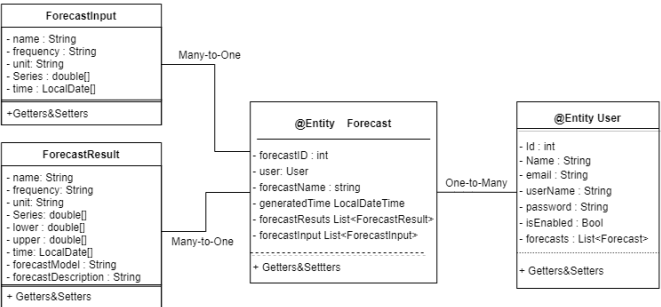
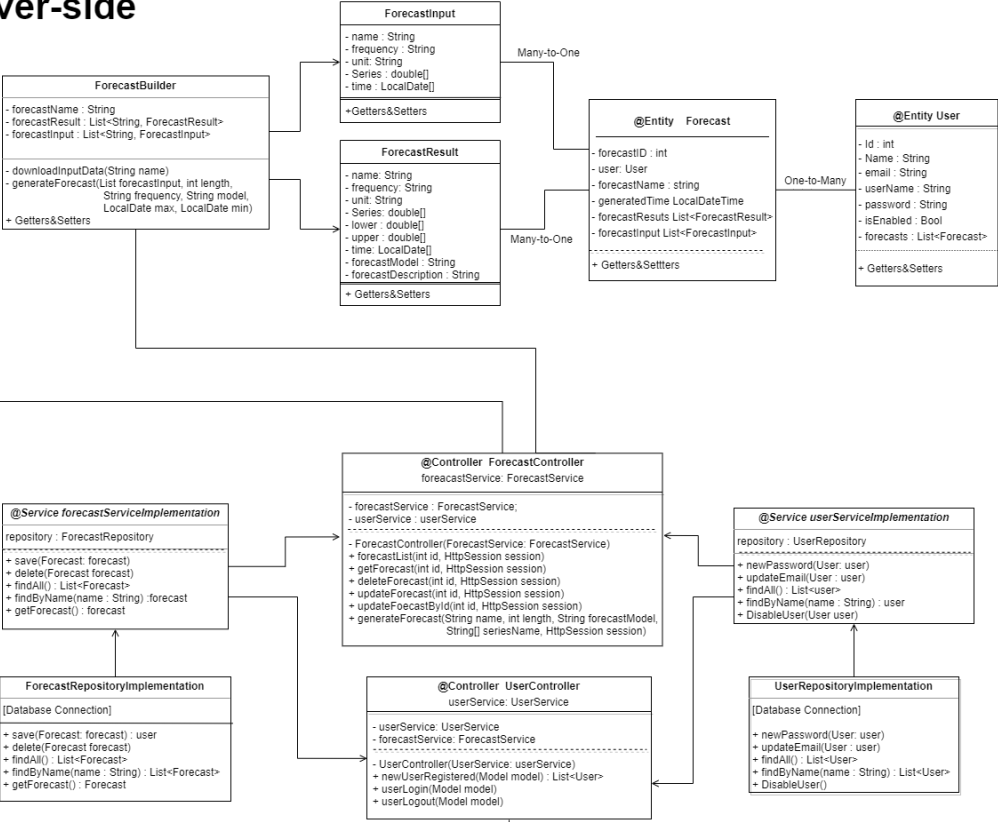
# Planned scope of initial release

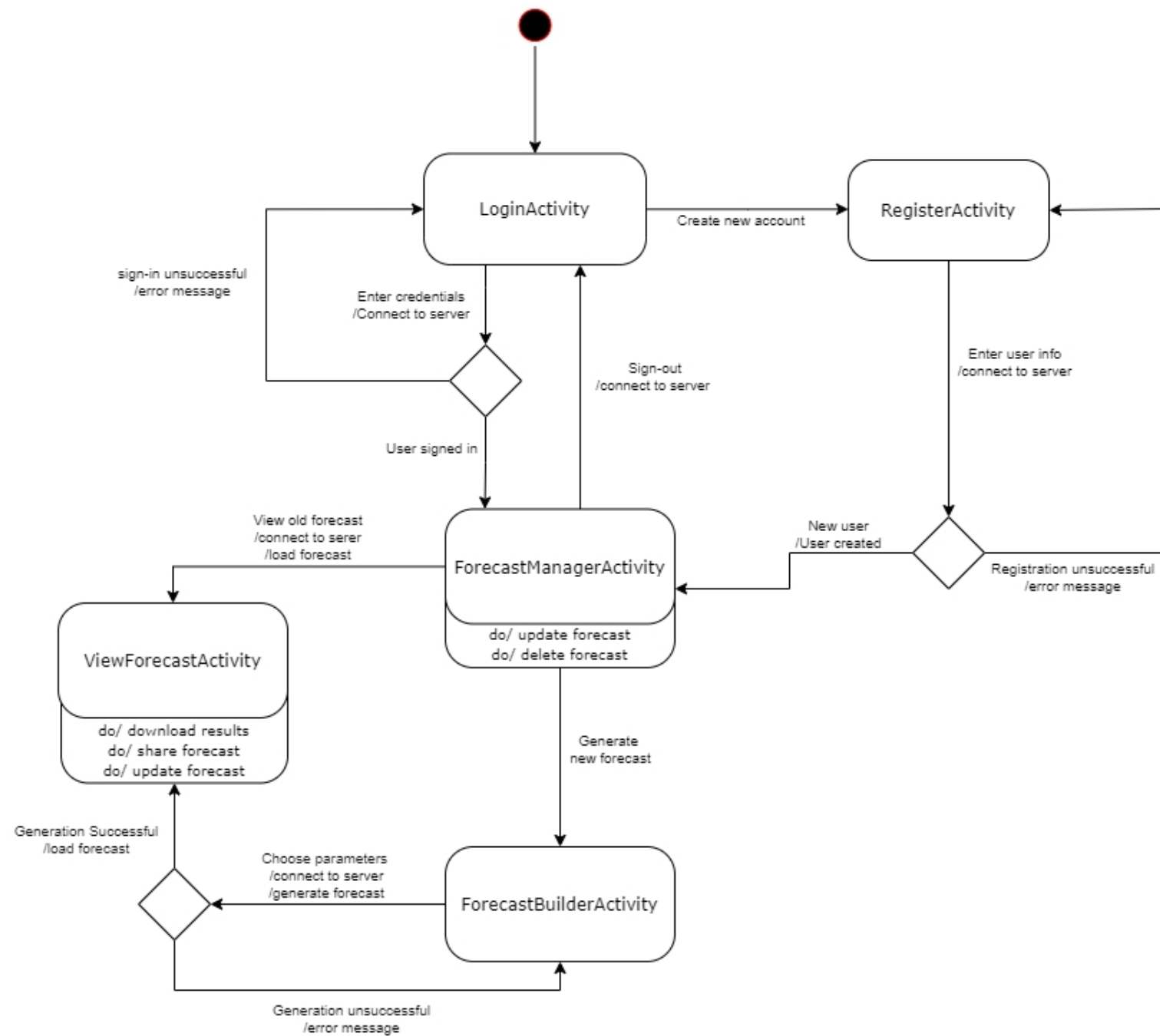
1. Users should be able to create accounts and then sign in
2. The app should be able to automatically retrieve data from Statistics Iceland and process it accordingly
3. The app should be able to create usable forecasts using the data from Statistics Iceland
4. The app should be able to present users with the best forecast estimates and any possible uncertainties in a clear and concise manner.
5. The app should be able to save the forecasts the user has created, retrieve them when needed and update them on request.

Client-side



Server-side





# Project plan

	Assignment	User story	Day start	Day end	Number of calendar days
	Login for Android	N1, N2, N9, E1	1.2.2021	20.2.2021	18
	Move backend of Forecasting to Android	N3, N4, N5	8.2.2021	1.3.2021	27
	Foundation for Android frontend	N6	8.2.2021	12.2.2021	4
	Basic features implemented <b>Milestone</b>		1.3.2012		
	Assignment 2: Design Model		20.2.2021	28.2.2021	8
	Add extra features to forecasting model	N5, N8	4.3.2021	16.3.2031	12
	Add charts to Android forecast view	N7	10.3.2021	23.3.2012	13
	Assignment 3: Code Review		7.3.2021	14.3.2021	7
	Make adjustments in accordance with code review		15.3.2021	25.3.2021	10
	Buffer / Extra time to deal with possible backlog and fix bugs	Possibly (N10, N11, N12)	25.3.2021	20.4.2021	26
	<b>Initial release</b> <b>Milestone</b>		20.4.2021		

Development and challenges



# Development

- Although reasonable at the time, were not able to follow project plan without delays
  - Covid + schoolwork
- Android programming to some extent needlessly complicated and took more time to build workable knowledge than expected
- Complex datastructures on backend meant moving data between the frontend and backend more complicated than expected.
- Soon became apparent that app would essentially just be a mobile port of the web app from last year
  - Forecasting economic variables doesn't really cater to any mobile specific features (camera, gyros etc.)
  - A possible idea was to add the ability to scribble notes by hand into charts but implementing this was not possible due to time constraints.

# Changes to class architecture during development

## Backend

- Created a new backend class `AndroidController` to handle all incoming requests from app
  - Controllers used in HBV501G not suitable and deemed better to gather handling of all Android requests into single class
- Added a new Entity type - `ForecastAndroid`
  - A Forecast object where the user attribute has been removed
  - Forecast object too complex for GSON to handle and cross-referencing between User objects and Forecast objects caused problems.
  - May not be most optimal approach since requires maintenance of two classes instead of one
  - However, both classes share `ForecastInput` and `ForecastResult` objects where most attributes are stored.
  - Possible benefit to this approach that now extra attributes only relevant to Android may be added to `ForecastAndroid` class without having to also add to `Forecast` class.

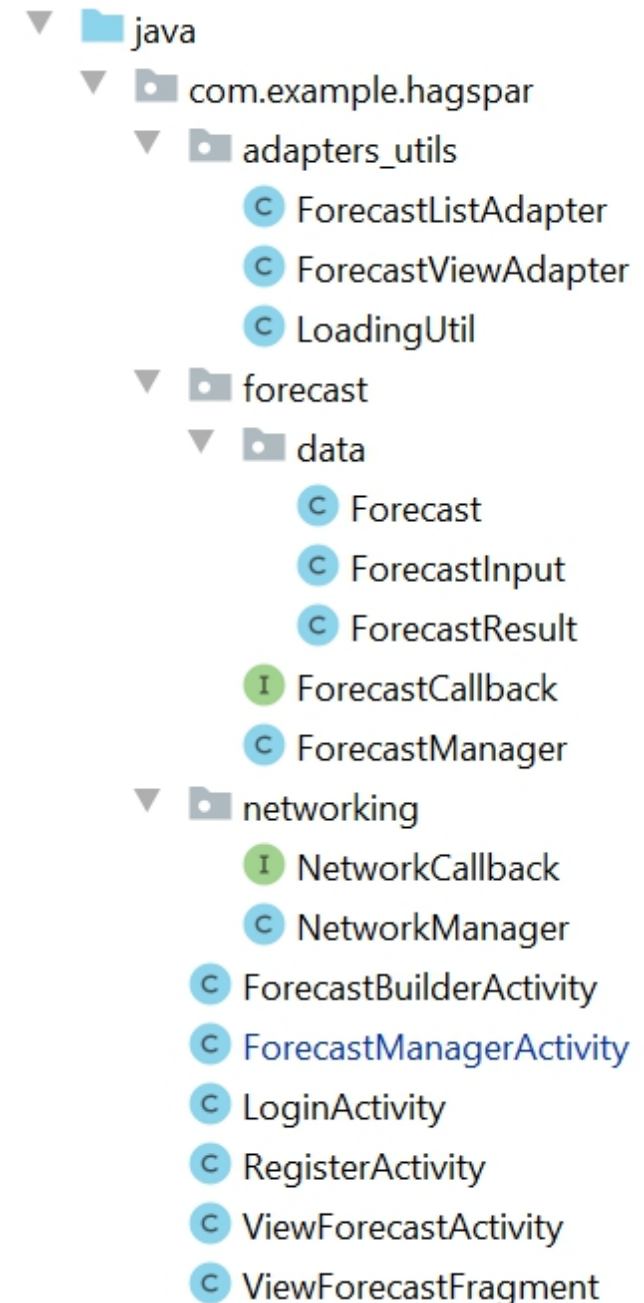
# Changes to class architecture during development

## Frontend

- Forecast class on the Android side was modified to be the same as backend ForecastAndroid class.
- User class and UserService removed from frontend
  - Not needed for initial release due to limited user management implementation
- Added NetworkManager class
  - Not fully aware of Android networking methods when class diagram created.
- ForecastService became ForecastManager
  - In order to better reflect its eventual implementation
  - Delivers and handles forecasts as planned, but also stores currently active forecast in memory (singleton with Forecast object as attribute).

# Eventual class structure

- Additional changes as compared to original class diagram included:
  - Unforeseen utility classes (two adapters and one to handle loading animations)
  - Callback interfaces to handle async networking
  - Fragment class to handle tabs in ViewForecastActivity



# Realized project plan

	Assignment	User story	Day start	Day end	Number of calendar days
	Login for Android	N1, N2, N9, E1	1.2.2021	<del>20.2.2021</del> 26.2.2021	<del>18</del> 24
	Move backend of Forecasting to Android	N3, N4, N5	<del>8.2.2021</del> 20.2.2021	<del>1.3.2021</del> 12.4.2021	<del>27</del> 51
	Foundation for Android frontend	N6	<del>8.2.2021</del> 20.2.2021	<del>12.2.2021</del> 10.3.2021	<del>4</del> 19
	Basic features implemented <b>Milestone</b>		<del>1.3.2021</del> 22.3.2021		
	Assignment 2: Design Model		20.2.2021	28.2.2021	8
	Add extra features to forecasting model	N5, N8	<del>4.3.2021</del> Never started	<del>16.3.2021</del>	<del>12</del>
	Add charts to Android forecast view	N7	<del>10.3.2021</del> 24.3.2021	<del>23.3.2021</del> 19.4.2021	<del>13</del> 27
	Assignment 3: Code Review		<del>7.3.2021</del> 25.3.2021	<del>14.3.2021</del> 31.3.2021	7
	Make adjustments in accordance with code review		<del>15.3.2021</del> Not applicable	25.3.2021	10
	<del>Buffer / Extra time to deal with possible backlog and fix bugs</del>	Possibly (N10, N11, N12)	<del>25.3.2021</del> All used for backlog	<del>20.4.2021</del>	<del>26</del>
	Initial release <b>Milestone</b>		<del>20.4.2021</del> 22.4.2021		

Initial release and lessons learned

# Realized scope of initial release

1. **Users should be able to create accounts and then sign in**
  - This was accomplished, albeit in a limited manner and lacking in actual security
2. **The app should be able to automatically retrieve data from Statistics Iceland and process it accordingly**
3. **The app should be able to create usable forecasts using the data from Statistics Iceland**
  - 2 and 3 were both accomplished, although the implementation was already in place (implemented as part of last year's project).
  - There were plans to improve the implementation and add additional features but in the end app development left no time for it.
4. **The app should be able to save the forecasts the user has created, retrieve them when needed and update them on request.**
  - This was fully accomplished and to this extent the app behaves exactly as the browser application from last year.
5. **The app should be able to present users with the best forecast estimates and any possible uncertainties in a clear and concise manner.**
  - This was partially accomplished. The user can view the forecasts results in charts which have some basic touchscreen features (zoom by pinching). However, other planned additional features (pop-ups on touch, additional styling) had to be left out due to time constraints.

# Lessons learned

- Not all applications should be ported to mobile
  - Added value for our application very limited
  - Most users would generate data and then use the data for further work, which is easy on a desktop machine but hard on mobile
- Include estimates for gathering knowledge and getting acquainted with new tools and methods in plans
  - Android programming proved a bigger hurdle than expected, despite it all being in a familiar language
- Importance of breaking activities into parts during planning
  - “Moving forecasting backend to Android” included to many tasks
  - Should have been broken down into 3-4 individual activities