

# Разработка инфраструктуры редактора диаграмм в браузере

Агеев Денис, 344 гр.  
СПбГУ, Математико-Механический факультет  
Научный руководитель: Тимофей Брыксин

# Введение

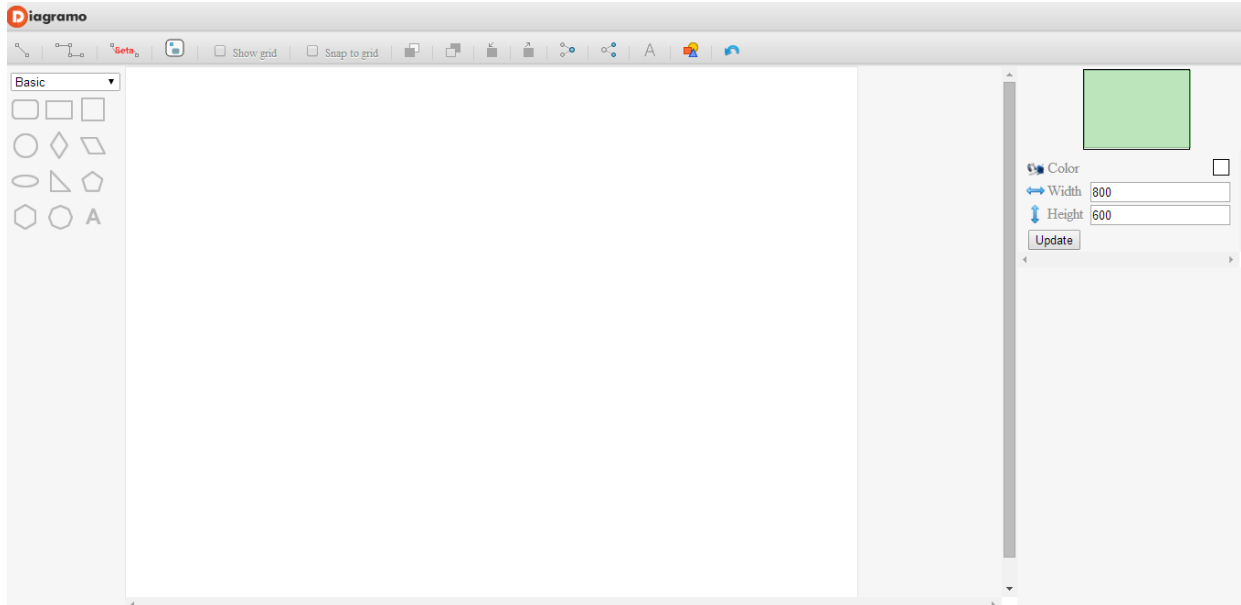
- Информационные технологии развиваются, рынок смартфонов и планшетов растет
- Есть потребность в создании множества приложений со сложной бизнес-логикой
- Нет решения, которое позволяло бы легко создать приложения со сложной логикой, не владея навыками программирования

# Постановка задачи

1. Проанализировать существующие решения редакторов диаграмм
2. Выбрать нужные инструменты для создания редактора
3. Разработать инфраструктуру редактора диаграмм, обладающей следующей функциональностью
  - a. возможность описания логики приложения с помощью диаграмм
  - b. легкая расширяемость редактора
  - c. импорт и экспорт диаграмм в формате JSON
  - d. Наличие таких компонент как палитра инструментов, рабочая область и редактор свойств

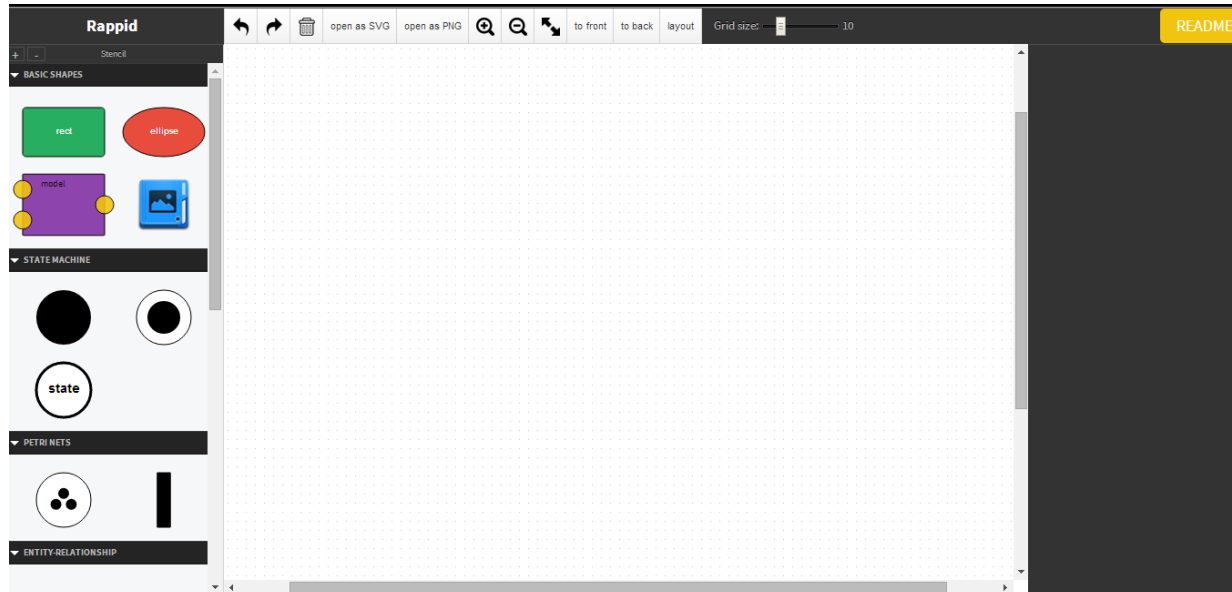
# Существующие решения

## Diagramo



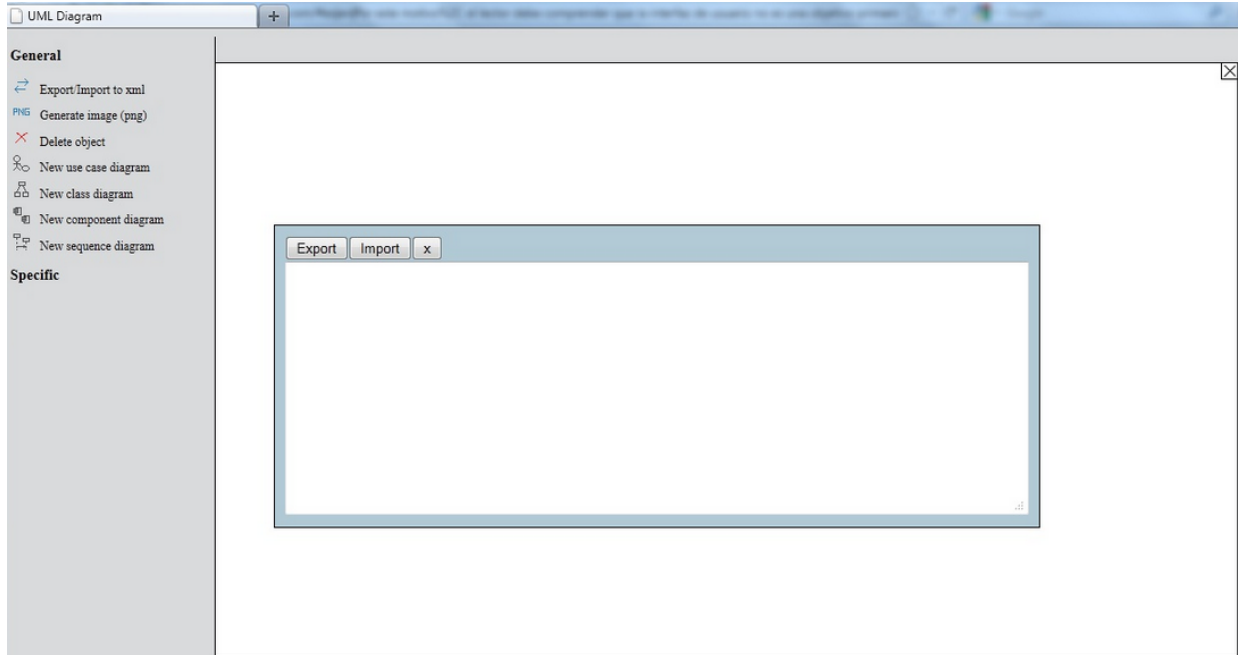
# Существующие решения

## Rappid



# Существующие решения

## jsUML2



# Инструменты

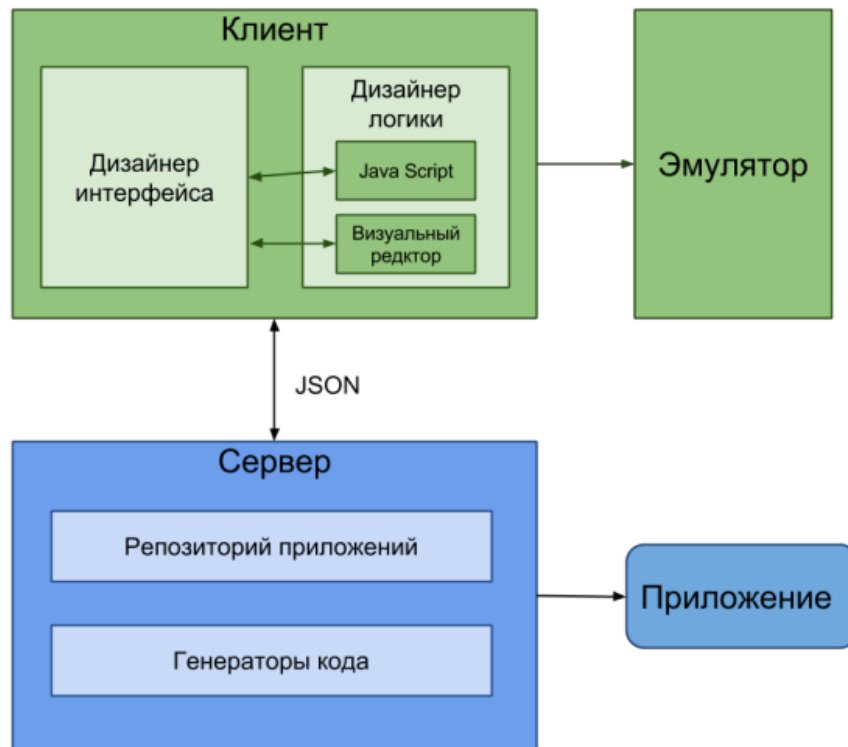
- TypeScript - основной язык разработки
- AngularJS как MVC Framework
- JQuery для удобной работы с DOM

# Сравнение JS библиотек для рисования диаграмм

|                           | Jgraph | Javascript<br>InfoVis ToolKit | Draw2D | JointJS |
|---------------------------|--------|-------------------------------|--------|---------|
| Бесплатна                 | -      | +                             | -      | +       |
| Хорошо<br>документирована | +      | +                             | +      | +       |
| HTML5 SVG                 | +      | -                             | +      | +       |
| Расширяемость             | +      | +                             | +      | +       |
| Проста в<br>использовании | +      | -                             | -      | +       |



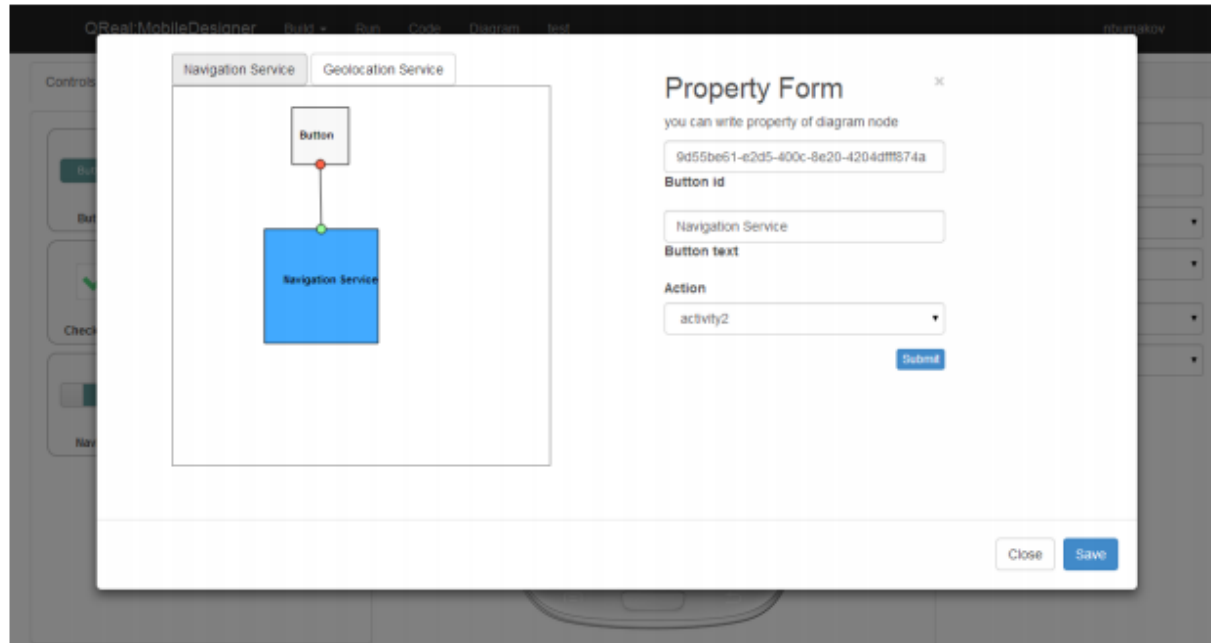
# Архитектура системы



# Архитектура редактора

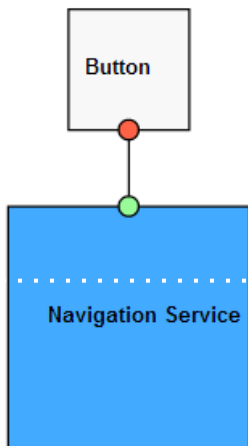


# Пользовательский интерфейс прототипа дизайнера логики

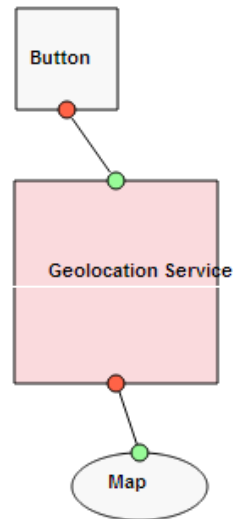


# Примеры сервисов

**Navigation Service** - служит для задания логики переходов между экранами



**Geolocation Service** - предназначен для задания логики работы с картами

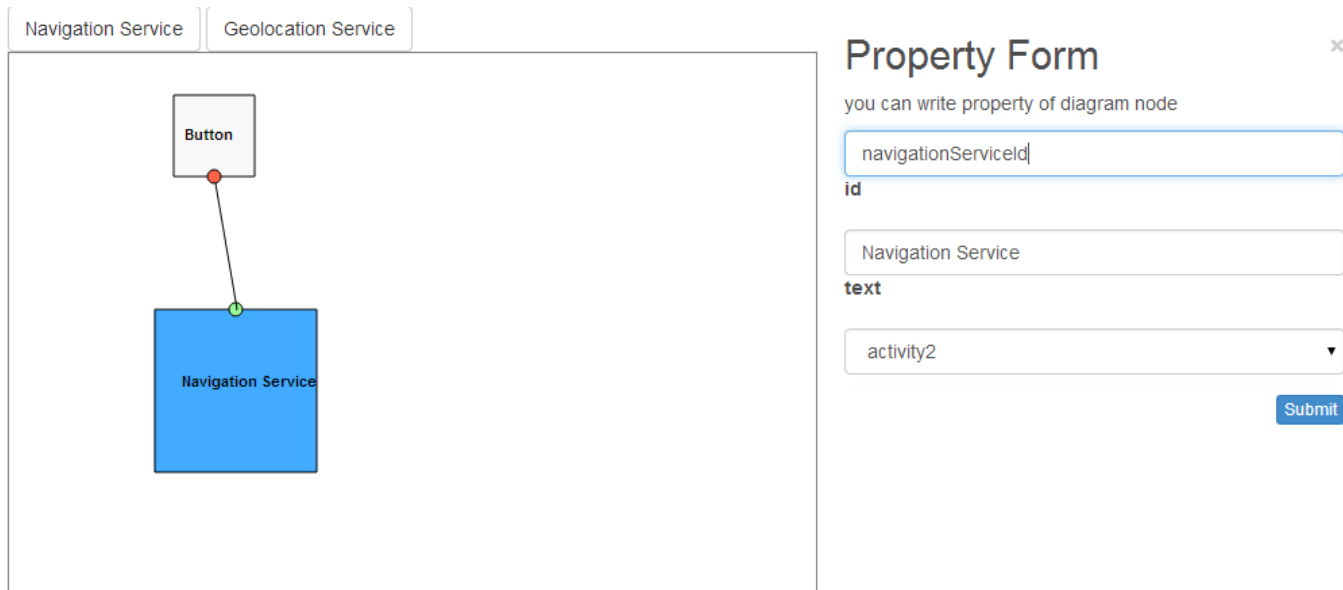


# Принцип работы

- Пользователь создает в конструкторе интерфейс приложения
- Нажимает на кнопку “задать логику”, открывается редактор логики
- Происходит импорт интерфейса приложения в редактор диаграмм и преобразование в диаграмму
- Пользователь задает логику приложений с помощью сервисов
- Жмет сохранить и получившаяся диаграмма в формате JSON отправляется в конструктор и далее на генератор приложений

# Апробация

## Приложение “Визитка”

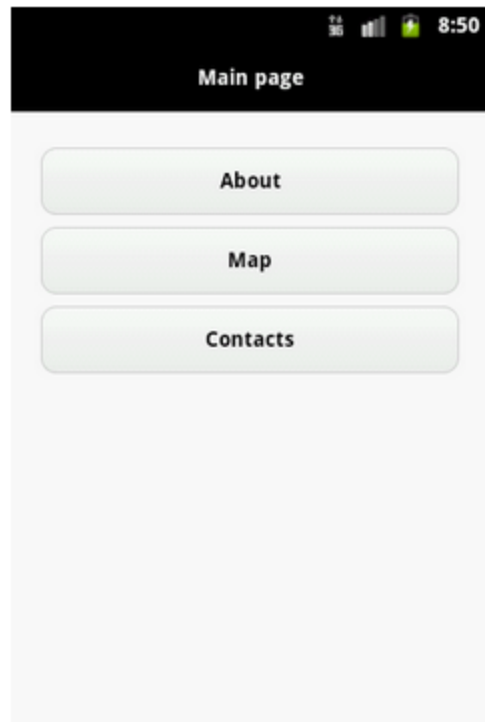
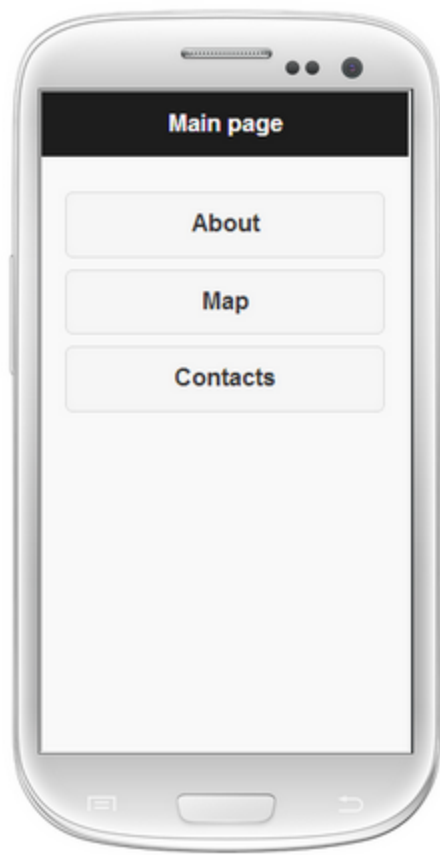


The image shows a software development environment with two main components: a diagram editor and a property form.

**Diagram Editor:** At the top, there are two tabs: "Navigation Service" (active) and "Geolocation Service". The diagram area contains a white box labeled "Button" at the top and a blue box labeled "Navigation Service" at the bottom. A line connects the bottom of the "Button" box to the top of the "Navigation Service" box, with a red dot at the top end and a green dot at the bottom end.

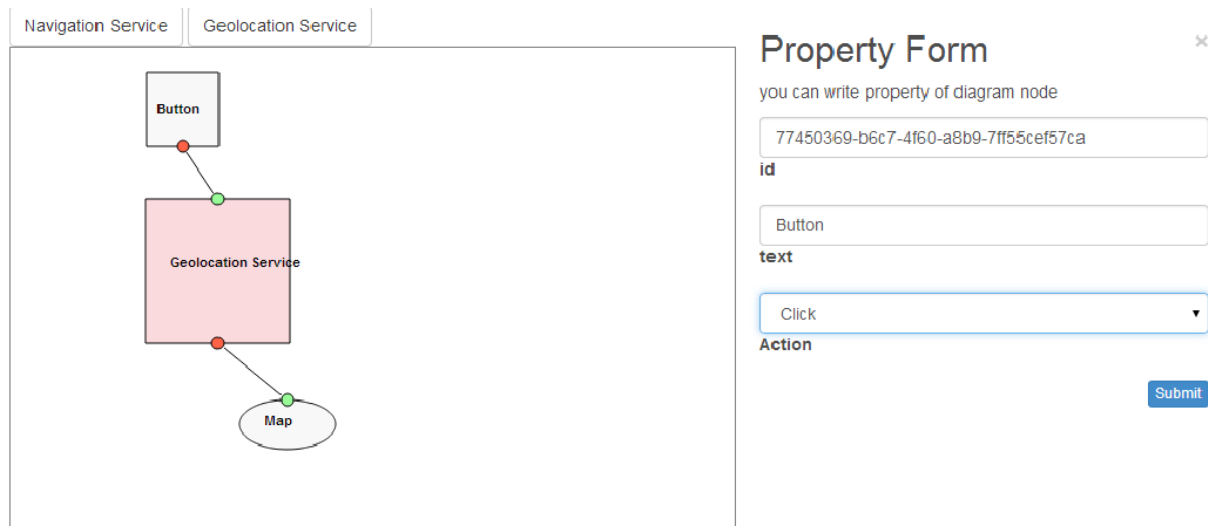
**Property Form:** A window titled "Property Form" is open on the right. It contains the following elements:

- A header: "Property Form" with a close button (x).
- A subtitle: "you can write property of diagram node".
- An input field for "id" containing the text "navigationServiceId".
- An input field for "text" containing the text "Navigation Service".
- A dropdown menu for "text" currently showing "activity2".
- A "Submit" button at the bottom right.

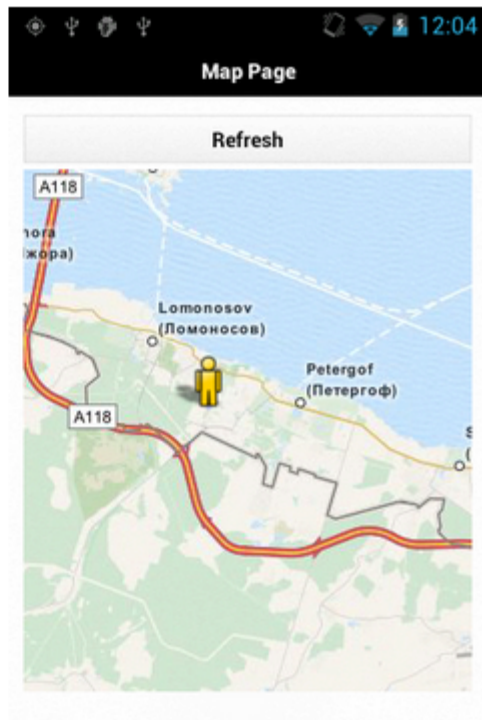
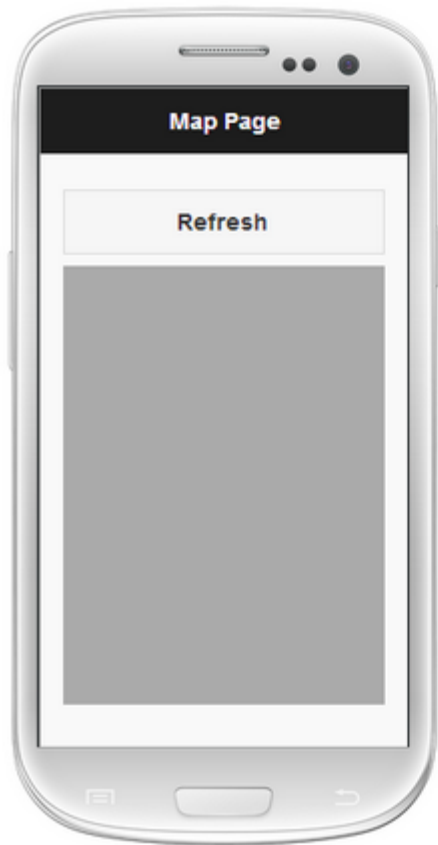


# Апробация

## Взаимодействие с сенсорами телефона







# Результаты

1. Проанализированы существующие решения редакторов диаграмм
2. Выбраны подходящие инструменты для создания редактора
3. Разработана инфраструктура редактора диаграмм со следующей функциональностью:
  - a. Поддержка описания логики приложения с помощью диаграмм
  - b. Возможность легкого расширения редактора, то есть простое добавление нужных элементов в наш метаязык
  - c. Возможность импорт и экспорта диаграмм в формате JSON
  - d. Валидация созданной диаграммы “на лету”
4. Были созданы такие компоненты как палитра инструментов, рабочая область и редактор свойств элементов
5. Проведена апробация редактора на конкретных примерах