**План тестирования**

**1. Введение**

**1.1. Цель**

Целью составления данного плана тестирования является описание процесса тестирования приложения «Судоку». Документ позволяет получить представление о мероприятиях по тестированию приложения.

**1.2. Состав документа**

Документ содержит описание общих для подсистем стратегии, подходов и видов тестов. Также определяет численные и квалификационные требования к персоналу, необходимые для успешного тестирования; необходимое программное и аппаратное обеспечение.

**1.3. Задача тестирования**

Задачей тестирования приложения «Судоку» является проверка его корректной работы в целом, кроме того, также должна быть протестированы отдельные модули, классы, методы и правильность взаимодействия модулей друг с другом. В случае обнаружения ситуаций, в которых поведение приложения является неправильным, нежелательным или не соответствующим спецификации, должно быть проведено повторное тестирование после исправления ошибок в реализации.

Итогом процесса тестирования должен стать развернутый обзор, подтверждающий выполнение всех требований, описанных в техническом задании, и, в случае дальнейшего сопровождения приложения, дающий разработчикам, уверенность в корректной работе данного приложения и его составных частей.

Проверка будет производиться с помощью CASE-средства jUnit по стратегии восходящего тестирования методами “чёрного” и “белого” ящиков.

**1.4. Требования к тестам**

Тесты должны:

1. Проверять соответствие приложения функциональным требованиям;
2. Проверять корректность работы  отдельных компонентов приложения;
3. Определить надежность работы приложения при заданных условиях (кол-во игроков онлайн и т.д.);

**2. Стратегия**

В процессе тестирования Android-приложения Судоку будет применено тестирование “белого” и “чёрного” ящиков. Тестирование методом «белого» ящика позволяет проверить внутреннюю структуру программы, а тестирование методом «черного» ящика позволяет проверить корректность выходных данных тестируемого модуля. В ходе тестирования черным ящиком будет использован прием эквивалентного разбиения, целью которого является сокращение числа тестов и полная покрываемость требований каждым модулем системы. В итоге данная техника должна привести к сокращению времени тестирования, а также улучшению структурированности тестирования.

На первом этапе будет произведено модульное тестирование, в рамках которого будут тестироваться основные методы и классы в целом. Предполагается, что такие базовые классы как ContextWrapper, Context, Service, ContentProvider.

На втором этапе будет проведено интеграционное тестирование, при котором будет использована восходящая стратегия так как она позволяет обнаружить ошибки на самых нижних уровнях программы, обеспечивает более простое наблюдение за результатами тестирования, а также позволяет легче создавать тестовые условия.

На третьем этапе будет проведено системное тестирование, основанное на корректном сценарии работы пользователя с приложением, описывающих полный цикл работы с приложением. В ходе системного тестирования планируется выявить несоответствия требованиям к системе и недостатки графического интерфейса.

На четвертом этапе будет проведено функциональное тестирование, в ходе которого будут проверены все функциональные возможности системы, указанные в требованиях на нетривиальных случаях.

На пятом этапе будет проведено конфигурационное тестирование, в ходе которого приложение будет протестировано на различных платформах с целью проверки выполнения нефункциональных требований у системе.

Основным объектом тестирования является класс Field, который представляет собой игровое поле. Данный класс предоставляет метод проверки заполненности поля, а также основной метод, реализующий алгоритм вычисления сложности заполненной пользователем клетки.

**2.1. Типы тестов**

**2.1.1. Модульное тестирование**

**Цель:**

Протестировать основные классы и методы системы.

***Описание процесса тестирования:***

Тестирование классов и методов системы с помощью case-средства JUnit.

***Критерий завершения тестирования:***

Все методы возвращают корректные значения.

**2.1.1. Системное тестирование**

***Цель:***

Убедиться, что пользователь может использовать основные функционалы программы в процессе реального использования.

Проверить степень удобства приложения для пользователя на основании общих критериев и субъективного восприятия.

***Описание процесса тестирования:***

Эмуляция полного цикла использования модуля: предоставление входных данных в соответствии с выбранным алгоритмом, получение оптимального решения в виде согласно п.3.5., визуализация полученных решений на карте Google, сохранения результатов в Системе и отдельным файлом Excel (с расширением .xlsx).

***Критерий завершения тестирования:***

Приложение корректно рассчитывает оптимальные маршруты инкассаций, сохраняет их в Системе и правильно отображает на карте Google.

**2.1.2. Функциональное тестирование**

***Цель:***

Выявить функциональные ошибки, основанные на взаимодействии функционалов путем реализации нетривиальных сценариев.

***Описание процесса тестирования:***

Производится условное разделение приложения на функционалы, на основании этого разделения проводится целенаправленное тестирование:

1. Вычисление маршрутов;
2. Визуализация;
3. Сохранение.

***Критерий завершения тестирования:***

1. Все запланированные области протестированы;
2. Все найденные ошибки и замечания зафиксированы.

**2.1.3. Конфигурационное тестирование**

***Цель:***

Проверить корректную работу приложения на различных операционных системах.

***Описание процесса тестирования:***

Проверка работоспособности модуля на следующих операционных системах:

1. Windows 8 — обязательно;
2. Windows 8.1 — обязательно;
3. Windows 10 — обязательно.

***Критерий завершения тестирования:***

Проверена корректность работы на отмеченных как обязательные операционных системах.