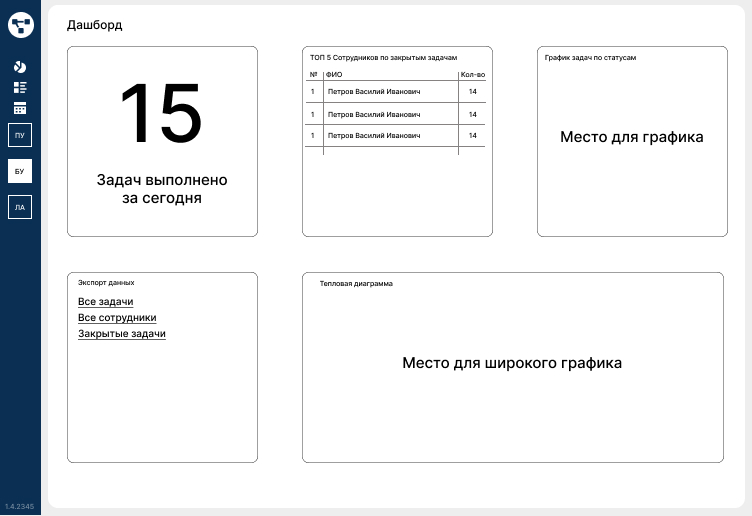
# Сессия 3

|  |  |
| --- | --- |
| **Наименование репозитория/ветки** | **Раздел задания** |
| Desktop/Project\_Management | Desktop-разработка |
| Library/Employee\_Time | Алгоритмы. Разработка библиотеки |
| Designing/UI | Проектирование UI |
| Designing/UML | Документирование |
| Test/Case | Тест-кейсы |

## Desktop-разработка

### Dashboard по проекту



Планируется, что данный раздел будет ключевым для руководства и постоянно открыт у определенных типов пользователей. Информация в блоках должна обновляться каждые 30 секунд.

Необходимо реализовать раздел с выгрузкой по данным в приложении. Данный разел должен быть собран из отдельных компонентов, которые должны быть размещены в интерфейсе в виде сетки для удобного отображения при разных размерах окна. Например,

* если пользователь просматривает окно на всю ширину - должны помещаться 4 элемента в одну линию,
* при уменьшении размера окна - 3 или 2 элемента в одну линию, а остальные должны переместиться ниже.

Таким образом пользователь должен всегда видеть информационные блоки в удобном режиме.

Блоки (не отступы между ними) должны растягиваться и сжиматься при изменении размеров окна (быть “резиновыми”) и только в случае критически маленького размера блоков они должны быть перемещены.

Дашборд должен формироваться по выбранному проекту и содержать описанные ниже блоки.

Пользователь должен иметь возможность располагать блоки дашборда в нужном порядке (приоритете) с помощью функции drag&drop для любого количества блоков в дашборде. При перетаскивании блока между двумя другими все последующие должны сдвигаться дальше от начала (снижать свой приоритет). Например, пользователь перетаскивает блок #7 на место блока #3: блоки с 3 и дальше должны сместиться на одну позицию, а блок #7 встать после второго. Расположение блоков, установленное пользователем, должно сохраняться на устройстве и после перезагрузки приложения.

* Блок с выводом количества незавершенных задач;
* Блок с выводом количества просроченных задач. В случае если просроченных задач больше двух, необходимо выделять блок красным цветом;

## Алгоритмы. Разработка библиотеки

Разработайте библиотеку расчета эффективности использования времени сотрудника с учетом коэффициента эффективности работы сотрудника, времени простоя, времени простоя без задач, времени простоя по вине другого сотрудника и прочих факторов.

Необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

Описание входных/выходных параметров, методику и примеры расчета, а также тестовые данные для тестирования библиотеки можно найти в ресурсах к заданию.

## Проектирование UI

Wireframes — это организация полного функционала конечного продукта, в виде структуры с представлением элементов интерфейса и навигации, их взаимодействия друг с другом. Как правило, в wireframes не используется типографика, цвет и любые другие графические элементы оформления, так как основное внимание уделяется функциональности, поведению и содержанию.

Разработайте wireframe для мобильного клиента. Мобильное приложение предназначено для клиентов. При разработке wireframe вам необходимо соблюдать принципы UX для дизайна мобильного программного обеспечения.

Ниже приведены некоторые требования:

* все необходимые поля должны быть функциональными (не должно быть «пустых» областей);
* кнопки и другие элементы управления должны быть четко обозначены;
* оптимальное использование полей ввода, списков, меню и т.д.

Разработайте  wireframe для мобильного приложения пользователя системы. Мобильное приложение ориентировано на исполнителя задач. В макете обязательно должны быть учтены экраны:

1. Список задач
2. Информация по задаче
3. Профиль пользователя
4. Чат с сотрудниками по проекту

## Тест-кейсы

Для выполнения процедуры тестирования библиотеки по расчету эффективности времени работы сотрудника . Вам нужно описать пять сценариев. Прецедент может быть выполнено успешно, а может быть отклонен согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон.

## Документирование

### Postman-коллекция

Необходимо сформировать postman-коллекцию для созданного вами проекта серверной части с описанием логики его работы, общей информации о проекте и отображением информации по каждому созданному запросу и примеры успешных и неуспешных выполнений. Наименование для коллекции: api\_doc\_{номер вашего рабочего места}.postman\_collection.json

### Диаграмма деятельности

Разработайте диаграмму деятельности для прецедента «Назначение задачи исполнителю». Сохраните файл с диаграммой в форматах .vsdx/.drawio и .pdf.

### Диаграмма последовательности

Разработайте диаграмму последовательности UML для описания процесса расчета заработной платы сотрудника. Алгоритм расчета представлен в описании библиотеки. Сохраните файл с диаграммой в форматах .vsdx/.drawio и .pdf.