**4** РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

## **4.1** Проектирование программ сверху вниз

Самое главное условие успешного создания как крупного, так и небольшого приложения заключается в применении надежных методов проектирования.

Самое широкое распространение при написании такого рода программ получили следующие методы:

* нисходящий (сверху-вниз);
* восходящий (снизу-вверх);
* специальный (на данный конкретный случай).

В случае нисходящего метода процесс разработки программы начинается с программы высокого уровня и спускается до программ низкого уровня.

В свою очередь восходящий метод работает в обратном направлении, то есть сначала идет разработка отдельных специальных подпрограмм, на основании которых в последствии строятся более сложные, которые заканчиваются самым верхним уровнем приложения.

Специальный подход не имеет какого-то заранее установленного способа разработки, то есть решения в выборе метода происходят во время процесса разработки и могут меняться в любой момент времени.

JavaScript является, в одно время, как структурированным так и не структурированным языком. Именно поэтому в качестве разработки был выбран язык TypeScript. Он представляет собой строгий и структурированный JavaScript и лучше всего подходит для нисходящего программирования. Нисходящий подход позволяет писать ясный, легко читаемый код, что в дальнейшем не вызовет проблем с сопровождением приложения. К тому же данный метод отлично подходит для прояснения и создания всей структуры программы в целом до кодирования более низких уровней приложения.

Выбранный метод позволяет уменьшить потери времени, обусловленные неудачными и ошибочными решениями на начальном этапе. А в условиях небольшого срока разработки программы – это делает его основным методом разработки и позволяет перейти к проектированию программы.

## **4.2** Структурирование программы

Как описывалось выше, для любой общей схемы при применении метода «сверху-вниз» разработку начинают с общего описания приложения, а после этого двигаются в направлении проработки ее конкретных деталей, компонентов и прочего.

При разработке любой программы лучше всего точно определить, что и как программа будет делать на самом высоком уровне, а затем уже погружаться в детали, касающиеся каждого ее действия. На рисунке 4.1 представлена схема нисходящего метода разработки.



Рисунок 4.1 – Схема нисходящего метода разработки программы

Исходя из изображения выше можно заметить, что эта схема очень легко ложиться на структурную схему разрабатываемого приложения, что говорит о правильном выборе метода разработки. Согласно схеме, можно легко определить и выделить два основных модуля программы. Модулей два потому как приложение содержит клиентскую и серверную части, каждая из которых служит обособленным модулем высокого уровня. К модулям высокого уровня относятся:

* блок ядра клиентского приложения;
* блок ядра серверного приложения.

Теперь можно приступать к составлению перечня действий, которые будет выполнять каждый отдельный модуль.

## **4.3** Блок ядра серверного приложения

Разработка начинается с ядра серверного приложения потому как разрабатывать клиентское ядро без минимально функционирующей серверной части не имеет смысла.

Первая и самая основная цель серверной стороны – это запускать сервер. Так как сервер будет работать на платформе NodeJS, то самый удобный и безопасный способ работать с сервером – это подключить библиотеку ExpressJS. Это уже давно зарекомендовавший себя модуль, который имеет большой и гибкий набор конфигураций.

Необходимо добавить данную библиотеку в проект с помощью NPM-менеджера. Для этого нужно использовать команду: «npm install express --save». После установки нужно подключить модуль к основному файлу сервера и произвести пробный запуск сервера на 1337 порту. По умолчанию, сервер будет запущен на локальном адресе компьютера без каких-либо дополнительных настроек. Пробный запуск сервера представлен на рисунке 4.2.



Рисунок 4.2 – Тестовый запуск сервера на ExpressJS модуле

После успешного тестового запуска сервера можно приступить к настройке его конфигураций.