

0.1 ОПИС ПРОГРАМНИХ ТЕХНОЛОГІЙ

Система поділяється на дві частини: сервер та клієнт.

Сервер реалізовано на мові PHP5. Це скриптова мова програмування, орієнтована на задачі web-розробки. Завдяки відсутності етапу компіляції зменшується час розробки та відлагодження. Динамічна типізація мови дозволяє прозоро працювати як з числами, так і з їх текстовим представленням, а також спрощує перевірку наявності об'єктів у булевих виразах. Мова надає вбудовані засоби для обробки рядків, параметрів HTTP-запитів та взаємодії з різними СКБД. Окрім цього, PHP є єдиною підтримуваною мовою на багатьох хостингах для web-сайтів та web-застосунків. З цієї ж причини у якості СКБД обрано MySQL 5. Задля використання нових можливостей, зокрема, скороченого оголошення масивів, мінімальною підтримуваною версією є PHP 5.4.

Клієнт реалізовано на скриптовій мові програмування ECMAScript 5 (більше відомій як JavaScript) та описових мовах HTML 5 та CSS 3. Вони є фактичними web-стандартами та підтримуються сучасними web-браузерами, такими як Firefox, Chrome, Edge, Safari, Internet Explorer, Android Browser тощо. Альтернативи не є кросбраузерними, приміром, мова VBScript підтримується лише у Internet Explorer, а мова Dart — лише у Chrome. Крім того, використання web-технологій дозволяє створювати на основі клієнту застосунки для настільних та мобільних комп'ютерів за допомогою спеціальних середовищ на основі браузерних рушіїв (напр., node-webkit, PhoneGap, AppJS).

0.2 ОПИС ПРОГРАМНИХ БІБЛІОТЕК

Оскільки клієнт є "товстим", на сервер покладені лише задачі передачі та обробки даних між клієнтом та базою даних, тож немає потреби у використанні систем керування контентом, фреймворків та тому подібних бібліотек. Проте мова PHP не має вбудованих засобів для побудови PDF-документів, тож для цієї задачі використано бібліотеку MPDF, яка формує документ з HTML-шаблону. Серед альтернатив розглядалася також бібліотека TCPDF, проте при спробі використання вона виявила надто погану підтримку можливостей HTML та CSS.

Модульні тести для серверу також написані мовою PHP та використовують бібліотеку Dogpatch, розширену та доповнену для потреб тестування системи. Бібліотека DogPatch дозволяє виконувати HTTP-запити та таким чином перевіряти, чи вірні відповіді та HTTP-коди надають AJAX-запити за певних умов. Додано функції для обробки відповідей у JSON, що дозволяють перевірити наявність певного об'єкту у масиві об'єктів.

Клієнт використовує бібліотеки jQuery та Kendo UI. jQuery є прошарком над API, що доступні для браузерного JavaScript, який надає компактний синтаксис.

ксист та кросбраузерні реалізації для таких частовживаних операцій, як робота з DOM та AJAX-запитами. Kendo UI надає елементи керування (віджети) із широкою функціональністю, у тому числі такі, що не надаються засобами HTML. Крім того, вона надає багатофункціональний елемент керування для представлення та редагування табличних даних, а також самостійно здійснює запити з цими даними до серверу, слід лише описати URL та формати даних. Для призначення гарячих клавіш використовується плагін для jQuery `jquery.hotkeys`.

0.3 ОСОБЛИВОСТІ СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ З УРАХУВАННЯМ МОВИ ПРОГРАМУВАННЯ

Інтерпретатор та мова PHP розраховані на використання у якості CGI, тобто web-сервер використовує URI як відносний шлях до файлу, запускає файл на виконання та віддає результат його роботи. Альтернативні способи використання URI, зокрема, маскування структури програмних модулів за допомогою так званих "людинозрозумілих URL" та маршрутизації запитів до викликів методів класів вимагають низькорівневих перехоплень за допомогою правил ModRewrite для Apache та аналогічних засобів для інших web-серверів. Тому натомість архітектуру серверної частини системи реалізовано без застосування об'єктно-орієнтованого програмування, проте із наслідуванням деяких його принципів. Модулі організовані по директоріях за концептуальними класами і реалізують інтерфейси шляхом розміщення у директоріях файлів з однаковими іменами, способами передачі параметрів та форматами вхідних та вихідних даних — це дозволяє за реалізації роботи з різними концептуальними класами у клієнті змінювати лише назву директорії. Функції для зв'язку між концептуальними класами винесено в окремі модулі, спільні функції винесено в окремі бібліотеки функцій.

Клієнт реалізовано як односторінковий застосунок. Головний інтерфейс системи реалізований HTML-сторінкою з меню; за натисненням пунктів меню створюються та відкриваються певні вікна засобами JavaScript. Проте сторінка входу та реєстрації, як незалежна від головного інтерфейсу сутність, винесена в окрему HTML-сторінку, до якої не підключено JavaScript-бібліотек. Обидві сторінки можуть змінюватися в залежності від режиму, повідомлень системи, прав доступу тощо, тому виводяться скриптами на мові PHP, що є HTML-шаблонами зі вставками коду на PHP. Власних функцій на JavaScript небагато і слугують вони здебільшого прошарком між описами даних та реалізованими засобами бібліотеки Kendo UI елементами керування, тож зібрані в один модуль. За розвитку системи та додання нових функцій може знадобитися деконпозиція цього модулю.

0.4 ОСОБЛИВОСТІ СТВОРЕННЯ СТРУКТУР ДАНИХ

У якості формату передачі даних від сервера до клієнту обрано JSON. Це текстовий формат на основі мови JavaScript, що надає компактний безнадлишковий (порівняно з похідними від SGML мовами) синтаксис та дозволяє структурувати рядки та числа у вкладених масивах та об'єктах (асоціативних масивах). JSON формується засобами мови PHP, що доступні з версії 5.1. Дані від клієнта до сервера передаються у тілі POST-запиту у форматі HTML-форм. Структури даних, що використовуються при роботі з бібліотекою Kendo UI, визначаються документацією до цієї бібліотеки.