

Практическая работа 1

Разработка сервиса предложения решений

```
import flet as ft
import sqlite3
from datetime import datetime

def init_db():
    conn = sqlite3.connect('proposals.db')
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS proposal
                    (id INTEGER PRIMARY KEY AUTOINCREMENT,
                     department TEXT,
                     proposal_text TEXT,
                     priority TEXT,
                     deadline TEXT,
                     cost REAL)")
    cursor.execute('SELECT COUNT(*) FROM proposal')
    if cursor.fetchone()[0] == 0:
        data = [
            ('Пивной отдел Пятерочки', 'Купить пиво', 'Высокий', '2026-03-01',
             150000),
        ]
        cursor.executemany('INSERT INTO proposal (department, proposal_text,
priority, deadline, cost) VALUES (?, ?, ?, ?, ?)', data)
    conn.commit()
    conn.close()

def get_proposals():
    conn = sqlite3.connect('proposals.db')
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM proposal')
    rows = cursor.fetchall()
    conn.close()
    return rows

def add_proposal(department, proposal_text, priority, deadline, cost):
    conn = sqlite3.connect('proposals.db')
    cursor = conn.cursor()
    print(department, proposal_text, priority, deadline, cost)
    print(cursor.execute('INSERT INTO proposal (department, proposal_text, priority,
deadline, cost) VALUES (?, ?, ?, ?, ?)',
                         (department, proposal_text, priority, deadline, float(cost)))))

    conn.commit()
    conn.close()
```

```

def main(page: ft.Page):
    page.title = 'Предложения о расширении ИС'
    page.horizontal_alignment = ft.CrossAxisAlignment.CENTER
    page.vertical_alignment = ft.MainAxisAlignment.CENTER

    init_db()

    table = ft.DataTable(columns=[
        ft DataColumn(ft.Text('Подразделение')),
        ft DataColumn(ft.Text('Предложение')),
        ft DataColumn(ft.Text('Приоритет')),
        ft DataColumn(ft.Text('Срок')),
        ft DataColumn(ft.Text('Стоимость'))
    ], rows=[])

    def load_table():
        table.rows = []
        for p in get_proposals():
            cost_val = float(p[5]) if p[5] else 0
            table.rows.append(ft DataRow(cells=[
                ft DataCell(ft.Text(p[1])),
                ft DataCell(ft.Text(p[2])),
                ft DataCell(ft.Text(p[3])),
                ft DataCell(ft.Text(p[4])),
                ft DataCell(ft.Text(f'{cost_val:.0f} ₽'))
            ]))
        page.update()

    def open_add_dialog(e):
        dep_input = ft.TextField(label='Подразделение')
        prop_input = ft.TextField(label='Предложение')
        prio_input = ft.Dropdown(label='Приоритет', options=[
            ft DropdownOption(key='Высокий', text='Высокий'),
            ft DropdownOption(key='Средний', text='Средний'),
            ft DropdownOption(key='Низкий', text='Низкий')
        ])
        date_input = ft.TextField(label='Срок реализации (YYYY-MM-DD)')
        cost_input = ft.TextField(label='Стоимость')

    def save(e):
        if dep_input.value and prop_input.value and prio_input.value and
date_input.value and cost_input.value:
            add_proposal(dep_input.value, prop_input.value, prio_input.value,
date_input.value, cost_input.value)

```

```

load_table()
page.close(dlg)
page.update()

dlg = ft.AlertDialog(
    title=ft.Text('Добавить предложение'),
    content=ft.Column([dep_input, prop_input, prio_input, date_input,
cost_input], tight=True),
    actions=[ft.Button('Сохранить', on_click=save), ft.Button('Отмена',
on_click=lambda e: page.close(dlg))]
)
page.open(dlg)

def open_report_dialog(e):
    proposals = get_proposals()
    total_cost = sum(float(p[5]) if p[5] else 0 for p in proposals)
    report_date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    report_cols = [
        ft DataColumn(ft.Text('Подразделение')),
        ft DataColumn(ft.Text('Предложение')),
        ft DataColumn(ft.Text('Приоритет')),
        ft DataColumn(ft.Text('Срок')),
        ft DataColumn(ft.Text('Стоимость'))
    ]
    report_rows = []
    for p in proposals:
        cost_val = float(p[5]) if p[5] else 0
        report_rows.append(ft DataRow(cells=[
            ft DataCell(ft.Text(p[1])),
            ft DataCell(ft.Text(p[2])),
            ft DataCell(ft.Text(p[3])),
            ft DataCell(ft.Text(p[4])),
            ft DataCell(ft.Text(f'{cost_val:.0f} ₽'))
        ]))

    report_table = ft.DataTable(columns=report_cols, rows=report_rows)

    def print_report(e):
        print(f'Отчет сформирован: {report_date}, Всего предложений:
{len(proposals)}, Общая стоимость: {total_cost}')

    dlg = ft.AlertDialog(
        title=ft.Text('Отчет'),
        content=ft.Column([

```

```

        ft.Text(f'Дата формирования: {report_date}'),
        ft.Text(f'Всего предложений: {len(proposals)}'),
        ft.Text(f'Общая стоимость: {total_cost:.0f} ₽'),
        report_table,
        ft.Button('Распечатать', on_click=print_report)
    ], tight=True)
)
page.open(dlg)

def exit_app(e):
    pass

load_table()

page.add(ft.Column([
    ft.Text('Предложения о расширении ИС', size=24,
weight=ft.FontWeight.BOLD),
    table,
    ft.Row([
        ft.Button('Добавить предложение', on_click=open_add_dialog),
        ft.Button('Сформировать отчет', on_click=open_report_dialog),
        ft.Button('Выход', on_click=exit_app)
    ])
]))
if __name__ == "__main__":
    ft.app(target=main)

```

Работа системы показана на рисунке 1:

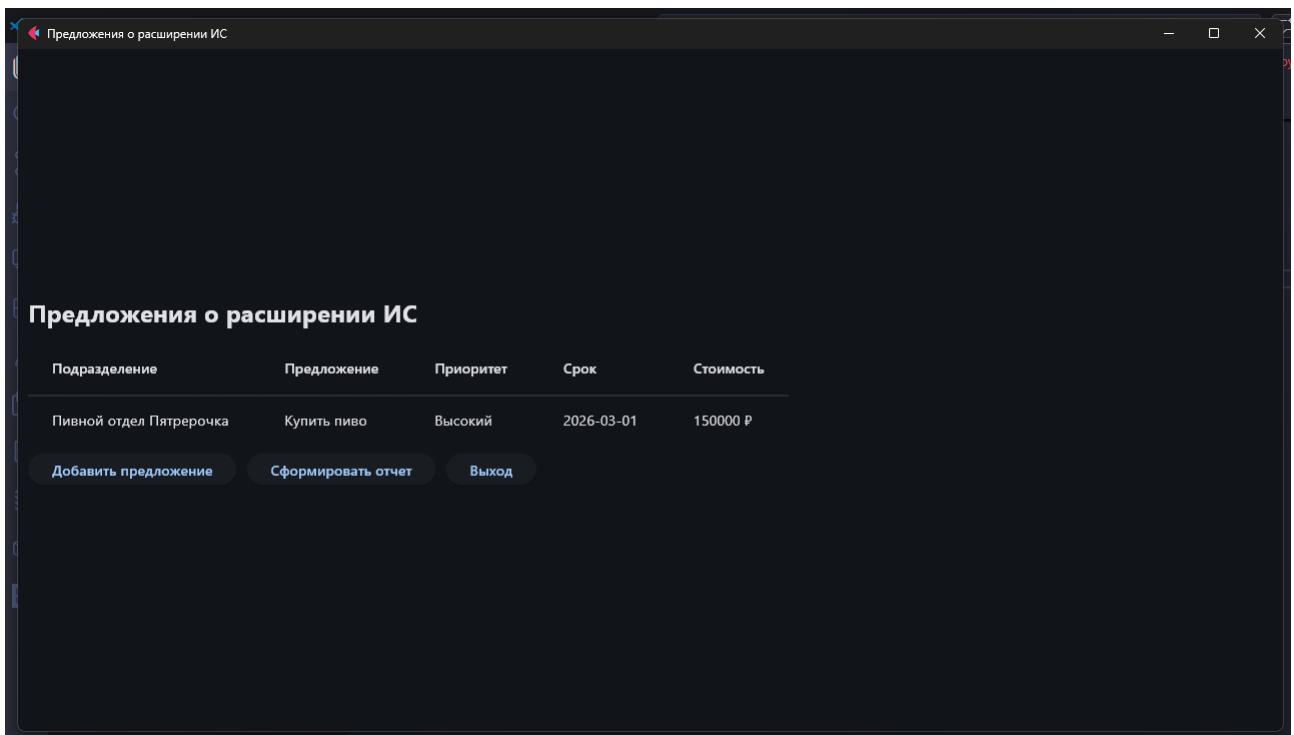


Рисунок 1 — Работа приложения